

PONTIFICA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**“SISTEMA DE ADMINISTRACIÓN DE BENEFICIOS
PARA DEPARTAMENTO DE BIENESTAR PUCV”**

LUIS ALEJANDRO ESPINOZA TAPIA

INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO DE EJECUCIÓN EN INFORMÁTICA

Enero de 2014

AGRADECIMIENTOS

a toda persona que ha sido de apoyo en el transcurso de esta etapa tanto amigos, familiares y compañeros de trabajo en especial a mi madre, padre, tía y funcionarios que he conocido dentro de Recursos Humanos en la PUCV.

Resumen

Dentro del Departamento de Bienestar PUCV existe un sistema informático desarrollado hace más de 25 años el cual aun siendo funcional no cumple con requerimientos surgidos en la actualidad y con lo que se espera de un sistema actual.

La solución que se plantea es la creación de un sistema en base a tecnologías actuales, incorporando a las funcionalidades del sistema con que se cuenta nuevos requerimientos. El desarrollo contempla la captura de los requerimientos en base a las funcionalidades del sistema actual para luego obtener a partir de los procedimientos del Bienestar los nuevos requerimientos incorporando al sistema los requerimientos de la Oficina de Beneficios a las Personas, lo que resulta en una consolidación de información en un solo sistema para dos secciones de la PUCV. Luego del análisis se opta por la arquitectura de 3 capas cliente-servidor, donde el lenguaje de programación en la capa lógica y del cliente es Java y el motor de base de datos seleccionado es MySQL. En una etapa siguiente se realizan prototipos para validar los requerimientos, lo cual culmina con una migración de datos.

La culminación del proyecto da como resultado la experiencia del trabajo con usuarios finales y la constante resistencia al cambio que se puede generar, siendo mitigada por la inclusión de estos a lo largo del desarrollo del proyecto.

Palabras-clave: iteraciones, prototipos, requerimientos, Universidad.

Abstract

There is a computer system, developed over 25 years ago which still remain functional does not meet requirements arising today and what is expected of a current system within the Department of Bienestar PUCV.

The solution that is proposed is the creation of a system using current technologies, incorporating the functionalities of the system with new requirements that account. The development provides for the capture of requirements based on the features of the current system and then get from Bienestar procedures incorporating the new requirements to the system requirements Benefits Office for People, resulting in a consolidation of information in one system for two sections of the PUCV. After the analysis is chosen architecture client server with three-layer, where the programming language in the logic layer and client is Java and selected database is MySQL. In a next step prototypes are built to validate the requirements, which culminates with a data migration.

The culmination of the project results in the experience of working with end users and constant resistance to change that can be generated, being mitigated by the inclusion of these throughout the project development.

Keywords: iterations, prototypes, requirements, university.

Índice

1	<i>Presentación del Proyecto</i>	2
1.1	<i>Introducción</i>	2
1.2	<i>Definición de la Problemática</i>	3
2	<i>Definición de Objetivos</i>	4
2.1	Objetivo General	4
2.2	Objetivos Específicos	4
3	<i>Desarrollo del Proyecto</i>	5
3.1	Estudio de Factibilidad	5
3.1.1	Factibilidad Técnica	5
3.1.2	Factibilidad Económica	5
3.1.3	Factibilidad Operacional	6
3.1.4	Factibilidad Legal	6
3.2	Paradigma de Desarrollo	7
3.2.1	Paradigma Iterativo	7
4	<i>Análisis Primera Iteración</i>	8
4.1	Requerimientos	8
4.1.1	Requerimientos Funcionales	8
4.1.2	Requerimientos No Funcionales	9
4.1.3	Definición de Actores	9
4.2	Diagramas de Casos de Uso	10
4.2.1	Caso de Uso General	10
4.2.2	Casos de Uso Específicos	11
5	<i>Diseño Primera Iteración</i>	15
5.1	Diagramas de Secuencia	15
5.2	Diagrama de Clase	17
5.3	Modelo de Datos	18
6	<i>Análisis Segunda Iteración</i>	19
6.1	Requerimientos	19
6.1.1	Requerimientos Funcionales	19
6.2	Diagramas de Casos de Uso	20
6.2.1	Caso de Uso General	20
6.2.2	Casos de Usos Específicos	21
7	<i>Diseño Segunda Iteración</i>	22

7.1	Diagrama de Clase.....	22
7.2	Modelo de Datos	23
8	<i>Arquitectura de la aplicación</i>	24
8.1	Arquitectura Cliente-Servidor de tres capas.....	24
9	<i>Implementación</i>	25
10	<i>Plan de Pruebas</i>	27
10.1	Tipo de Pruebas Utilizadas	27
11	<i>Migración de Datos</i>	28
11.1	<i>Método a Utilizar para la Carga Inicial</i>	28
12	<i>Conclusión</i>	29
13	<i>Referencias</i>	30
14	<i>Anexos</i>	31

Lista de Figuras

Figura 4.1 Caso de Uso General Primera Iteración	10
Figura 4.2 Gestionar Solicitudes	11
Figura 4.3 Gestionar Comprobantes	11
Figura 4.4 Generar Reportes y Contabilidad.	12
Figura 4.5 Gestionar Préstamos	12
Figura 4.6 Gestionar Proveedores.....	13
Figura 4.7 Gestionar Cuentas.....	13
Figura 4.8 Administrar Usuarios.	14
Figura 4.9 Gestionar Socios y Cargas.....	14
Figura 5.1 Diagrama de Secuencia Ingresar Socio	15
Figura 5.2 Diagrama de Secuencia Ingresar Solicitud	16
Figura 5.3 Diagrama de Secuencia Generar Comprobantes de Solicitudes	16
Figura 5.4 Diagrama de Clase	17
Figura 5.5 Modelo de Datos Primera Iteración	18
Figura 6.1 Caso de Uso General Segunda Iteración	20
Figura 6.2 Caso de Uso Gestionar Empleado	21
Figura 6.3 Caso de Uso Gestionar Seguro Catastrófico	21
Figura 7.1 Diagrama de Clase Segunda Iteración.....	22
Figura 7.2 Modelo de Datos Segunda Iteración	23
Figura 8.1 Arquitectura de la Aplicación	24
Figura 9.1: Sección relativa a USUARIOS.....	38
Figura 9.2: Sección relativa a PERSONA	38
Figura 9.3: Sección relativa a PROVEEDOR.....	39
Figura 9.4: Sección relativa a SOLICITUD.....	39

Figura 9.5: Sección relativa a COMPROBANTE.....	40
Figura 9.6: Sección relativa a CUENTAS CONTABLES.....	40
Figura 9.7: Sección relativa a PERSONA.....	41
Figura 9.8: Sección relativa a SEGURO CATASTROFICO.....	41

Lista de Tablas

Tabla 9.1 Caso de uso Ingresar Solicitud.....	31
Tabla 9.2 Caso de uso Calcular Beneficios.....	31
Tabla 9.3 Caso de uso Generar Comprobantes de Solicitudes.....	32
Tabla 9.4 Caso de uso Actualizar Deuda.....	33
Tabla 9.5 Caso de uso Buscar Comprobante.	33
Tabla 9.6 Caso de uso Ingresar Comprobante.	34
Tabla 9.7 Caso de uso Consultar Movimientos Socio.	34
Tabla 9.8 Caso de uso Actualizar Arancel.	35
Tabla 9.9 Caso de uso Consultar Libro.....	35
Tabla 9.10 Caso de uso Generar Balance.	36
Tabla 9.11 Caso de uso Listar Deudores.	36
Tabla 9.12 Caso de uso Ingresar Préstamo.	37
Tabla 9.13 Caso de uso Imprimir Préstamos.	38
Tabla 9.14 Caso de uso Imprimir Préstamos.	38
Tabla 9.15 Caso de uso Buscar Proveedor.	39
Tabla 9.16 Plan de pruebas	44

1 Presentación del Proyecto

1.1 Introducción

Proyecto centrado en la creación de un Sistema para la administración de beneficios del Departamento de Bienestar de la Pontificia Universidad Católica de Valparaíso. El cual es el encargado de propender al mejoramiento de las condiciones de vida y trabajo, como asimismo al cuidado y fomento de la salud del personal asociado a él.

La problemática que se abordará en este proyecto es la antigüedad del sistema con el que se cuenta actualmente, ya que posee un poco grado de automatización de procesos, exceso de redundancia de datos y gráficamente una interfaz poco intuitiva para un usuario ajeno al área contable. Dicho proyecto considerará en su estructura el manejo de los siguientes servicios que se prestan actualmente:

Cálculo de reembolsos por prestaciones médicas, el cual puede constar de una o varias consultas médicas y, o exámenes. Reembolsos por programas médicos, lo que se puede desglosar como tratamientos o intervenciones de cualquier tipo.

Préstamos otorgados por medio de cartas de garantía, lo que implica la transferencia de la deuda al Bienestar ya que es cancelado por dicho departamento, previo a la facturación del servicio.

Manejo de convenios con empresas relativas al área de la salud, las cuales incluyen farmacias, ópticas, clínicas y centros hospitalarios.

Contabilidad general, lo que implica registros contables asociados a cada uno de los movimientos del departamento, generando anual y mensualmente un balance contable así como reportes cuando se desee, para un apoyo a la hora de toma de decisiones en el área de gestión.

Por último el manejo de la información de todos los asociados y sus respectivas cargas médicas. Consolidando toda la información que actualmente se encuentra dispersa y de manera poco óptica de consultar, en un único sistema de información, lo cual conllevará una migración de datos, para poseer datos tanto actuales como históricos de dicho departamento.

En una primera etapa se realiza una descripción general del proyecto y se establecen los objetivos, seguido de eso un estudio de factibilidad del proyecto tanto técnica como económica, operacional y legal, así como la elección de la metodología ideal para llevar a cabo el desarrollo del proyecto.

En una segunda etapa se efectuará un análisis, considerando requerimientos tanto funcionales como no funcionales, como a su vez la captura de los actores que involucra el sistema y los diagramas de caso de uso de los conllevan.

Por último, el diseño, lo que involucra el diagrama de clase asociado a la solución y diagrama de secuencia, como también la arquitectura del sistema y su modelo de datos respectivo, lo que se da a conocer como arquitectura lógica de la base de datos a utilizar.

1.2 Definición de la Problemática

En la Dirección de Recursos Humanos de la Pontificia Universidad Católica de Valparaíso, existe el Departamento de Bienestar del Personal, el cual consta con un sistema informático de gran funcionalidad, pero que debido a su antigüedad, nuevos requerimientos de parte de los usuarios, poca automatización de ciertos procesos y ser un programa monousuario, se requiere una nueva herramientas informática que cumpla con los requerimientos de esta época. A continuación se detallara el proceso principal del sistema el cual corresponde al reembolso de beneficios por prestaciones médicas.

El sistema está dirigido al personal que en forma voluntaria acepte las políticas del Bienestar, de reembolsos por prestaciones a cambio de aportes monetarios mensuales, los cuales en adelante serán individualizados como Socios.

El proceso se inicia cuando un Socio presenta una solicitud por reembolso, junto a documentación de respaldo, en esta primera etapa se debe digitar los datos de la solicitud, lo cual al ser un sistema mono-usuario esta tarea se realiza en un periodo de tiempo extenso, en comparación a como lo haría uno multi-usuario.

En una segunda etapa se dimensiona la vulnerabilidad del sistema, luego de ingresar todas las solicitudes, se guardan en un diskette, para ser trasladado a un segundo computador donde se realiza la comprobación atreves de la contabilidad, de alguna deuda que pueda tener el Socio con el Bienestar, condonando deuda por reembolso o cancela la totalidad de la solicitud en caso que sea necesario. Este movimiento de información involucra duplicado de esta, y grados distintos de actualización en los equipos involucrados, además del riesgo de pérdida de los datos.

Podemos reconocer una tercera etapa, la cual involucra la impresión de cheques para proveedores y para socios por reembolsos, esta funcionalidad se implementó en un periodo de tiempo posterior al sistema central por lo que no funciona conjuntamente, el usuario debe ingresar a otro programa que lee la información de los archivos del sistema central, a lo cual se debe ingresar la asignación de la numeración de cheques entre muchos otros parámetros más, tornándose un proceso poco intuitivo para personal ajeno, así como también las interfaces graficas que posee tanto el sistema central como el subsistema de impresión de cheques.

2 Definición de Objetivos

2.1 Objetivo General

Desarrollo de un sistema para el manejo de beneficios y contabilidad en el departamento de BIENESTAR PUCV, que reemplace y mejore el sistema informático actual.

2.2 Objetivos Específicos

- Capturar los requerimientos iniciales en base a la funcionalidad del sistema actual con que cuenta el Bienestar.
- Estudio de los procedimientos del Departamento de Bienestar para capturar nuevos requerimientos no contemplados en el sistema actual.
- Desarrollo de prototipos para la validación de los requerimientos del Departamento de Bienestar.
- Implantar la solución en el Bienestar, en base a los prototipos antes desarrollados.
- Migrar los datos del sistema actual, para una marcha blanca con datos históricos, destinado al manejo de información de los Socio y sus posibles deudas con el Bienestar.

3 Desarrollo del Proyecto

3.1 Estudio de Factibilidad

El estudio de factibilidad permite dimensionar que tan probable es la realización del proyecto, para lo cual se especifican los siguientes aspectos:

3.1.1 Factibilidad Técnica

En el Departamento de Bienestar se presentan dos alternativas para la explotación del proyecto. La primera consiste en crear una arquitectura propia, con servidores ubicados en la Dirección de Recursos Humanos y utilizar la estructura de conectividad que ofrece la Pontificia Universidad Católica de Valparaíso la cual permite soportar la implementación de la solución propuesta.

La segunda alternativa consiste en alquilar o comprar un servidor a través de la Dirección de Informática y Comunicaciones y adoptar las normativas y restricciones que esta dirección imponga.

Por decisión en conjunto con diversas autoridades de la Dirección de Recursos Humanos y Departamento de Bienestar se opta por la segunda opción, debido principalmente a lo que significa el uso no compartido de un servidor.

3.1.2 Factibilidad Económica

El equipo de desarrollo está enrolado en la institución que requiere la elaboración del software y además se cuenta con la infraestructura tecnológica para la implementación, por lo que la inversión inicial es mínima, ya que recae el costo de tiempo de personal solamente sobre el Estudiante en práctica Luis Espinoza Tapia, el cual recibe una remuneración de \$150.000 mensuales. En cuanto al costo de Hardware, se verá reflejado en la compra de un servidor; por el lado del desarrollo, se trabajará con hardware facilitado por la organización. Las Características mínimas estimadas del servidor son:

Procesador: Equipo con procesador de 2.0 GHz x64

Memoria: 2 GB de RAM

Disco Duro: 500GB con 5GB libres en primera instancia para el montaje, lo cual por uso requerirá más espacio para los datos al cabo de un tiempo.

Para estas características se estima un costo asociado de \$800.000.

Además se optará por el desarrollo y explotación de licencias de tipo Pública General (GNU GPL) tanto para el sistema operativo como para las herramientas utilizadas.

Dada la envergadura de lo que se plantea para el nuevo sistema y su posible evolución dentro de la organización, los beneficios superan ampliamente los costos fijos de la implementación de un nuevo sistema, tanto en el ámbito de costo hora-hombre, como en la estandarización de procesos entre Departamentos de la Universidad y reduciendo drásticamente el tiempo de dichos procesos.

3.1.3 Factibilidad Operacional

El desarrollo de este sistema ha sido solicitado por el auspiciador corporativo, quien ha decidido autorizar el uso de esta nueva herramienta dentro de la organización. Además existe un considerable interés de parte de los actores internos del Departamento de Bienestar por contar con esta herramienta que permitirá facilitar el control de administración de beneficios asociado a socios. En cuanto a la usabilidad dependerá de las características de las interfaces del sistema, que deberán estar bien adecuadas a los requerimientos de usuario. Se destaca que ciertas características del sistema actual tanto funcionales como de usabilidad se mantendrán en el nuevo sistema a desarrollar para reducir el impacto sobre los usuarios.

3.1.4 Factibilidad Legal

Se debe tomar en cuenta el trato delicado de los datos para la generación de contabilidad, ya que es algo que debe ser certificado por el contador en jefe del Departamento de Bienestar a fin de cumplir con las normas contables establecidas.

Otro aspecto relevante en el ámbito legal es la seguridad de la información, ya que se cuenta con datos que no deben estar al alcance de cualquier persona o usuario del sistema, de este modo dichos datos deben cumplir con el reglamento interno del Bienestar, lo cual es factible para el proyecto.

Por lo antes mencionado, el proyecto es factible en su totalidad, dando paso a las siguientes etapas de este, en un ambiente de análisis y desarrollo propicio.

3.2 Paradigma de Desarrollo

Para este proyecto se opta por el Paradigma Iterativo, el cual se describe a continuación junto a las razones por las que se seleccionó.

3.2.1 Paradigma Iterativo

Este tipo de desarrollo combina las ventajas del Modelo Cascada y Evolutivo, considerando aspectos tan importantes como que los requerimientos de sistemas siempre evolucionan en el transcurso de un proyecto, así como el trabajo sobre tareas tempranas en un desarrollo son re-hechas más a menudo de lo que se quisiera.

Este paradigma de desarrollo consiste principalmente en la captura de los requerimientos más importantes o críticos para el cliente, estableciendo prioridades en el desarrollo; para luego definir incrementos, en donde cada uno proporciona un subconjunto de funcionalidades de primera prioridad. Lo antes mencionado genera una retroalimentación con el cliente, donde puede apreciar tempranamente un primer producto, familiarizándose con los incrementos.

Los riesgos de falla al utilizar este desarrollo son relativamente bajos debido a la reiterada entrega de incrementos al cliente, por lo que las falencias del software se reducen en cada una de las iteraciones y otorga al cliente una percepción de mejora del producto final con respecto a la primera iteración.

Se escogió este tipo de desarrollo debido a su flexibilidad a la hora de captar requerimientos secundarios, ya que al trabajar sobre un sistema que se encuentra funcionando el nuevo sistema considera como requerimientos de primera prioridad las funcionalidades del actual sistema, dejando abierta la posibilidad de incorporar requerimientos de prioridad menor, para etapas más lejanas

4 Análisis Primera Iteración

4.1 Requerimientos

4.1.1 Requerimientos Funcionales

A continuación se describe los requerimientos funcionales definidos para el proyecto:

Atender Solicitud de Reembolso

Recibir e ingresar al sistema la solicitud de reembolso por prestaciones médicas asociada a un socio o una carga, considerando el tiempo de carencia de 4 meses para los socios de nuevo ingreso.

Generar Comprobante, tanto de egreso como de registros contables.

Asociar una solicitud a un comprobante ya sea de egreso, en el caso de no poseer deuda, o un registro contable, abonando a la deuda del socio.

Mantener Comprobante

Ingresar un comprobante de forma manual en el caso de un pago a un proveedor, modificar un comprobante existente en caso de un error o rechazar en caso de no corresponder el comprobante.

Mantenición de Socio

Ingresar un socio al sistema junto a los datos que sean requeridos y la posible modificación de los estados de este.

Mantenición de Carga

Ingresar al sistema a una persona como carga de un socio, contemplando una fecha de vencimiento de dicho beneficio, así como modificar sus datos cuando sea necesario.

Mantenición de Proveedores

Ingresar los datos de un proveedor al sistema o modificar los mismos en caso que sea necesario.

Impresión de Cheques

Para los comprobantes que correspondan a egresos, asociarlos a un número de cheque para poder realizar un seguimiento adecuado.

Mantenición de Cuentas Contables

Agregar o modificar datos de una cuenta contable, así como ejecutar un abono o un cargo a esta por todos los movimientos del sistema que lo requieran.

Actualización de Arancel de Fonasa

Se realiza una vez al año por los cambios en el valor de las prestaciones médicas.

Liquidar Deuda

Se debe totalizar la deuda de un socio cada cierto periodo de tiempo para pasar a descuento en OTP.

Calcular Beneficios

En todas las prestaciones presentadas dentro de una misma solicitud para poder obtener un posible total a pagar.

Administración de Prestamos.

Agregar, modificar o renegociar préstamos otorgados por el Departamento de Bienestar.

Generar Reportes

Generación de informes, destinados al área de gestión, en cierto periodo de tiempo indicado por el usuario. Los tipos de análisis necesarios son por:

Socio, indicando todos los movimientos asociados al antes mencionado.

Proveedor, destinado a identificar los gastos incurridos en este ámbito.

Totalización de los gastos incurridos a través de los pagos de beneficios.

Administrar Contabilidad

Actualizar los libros Contables por cada movimiento que se requiera registrar, así como validar cada movimiento para que el débito y el crédito sean iguales.

Control de Usuarios

El acceso al sistema debe estar restringido por el uso de claves asignadas a cada uno de los usuarios. Solo podrán ingresar al sistema las personas que estén registradas.

4.1.2 Requerimientos No Funcionales

Performance (Desempeño)

El sistema debe ser capaz de dar respuesta al acceso de todos los usuarios con tiempo de respuesta menor o igual a 5 segundos.

Disponibilidad

La disponibilidad es un aspecto relevante en el sistema, ya que se debe poseer información en todo momento para ejercer de forma rápida y fluida los procesos que se demandan.

Usabilidad y Operatividad

El sistema debe ser de fácil operación y que demande un bajo nivel de soporte de los usuarios del sistema.

4.1.3 Definición de Actores

Nombre Actor: Digitador

Perfil: Persona encargada de ingresar las solicitudes al sistema, así como los nuevos socios y cargas.

Nombre Actor: Contador

Perfil: Persona encargada de revisar la información ingresada en las solicitudes, gestionar los préstamos, las cuentas contables, los proveedores y los comprobantes generados a partir de las solicitudes.

Nombre Actor: Director de RR. HH.

Perfil: Persona encargada de la gestión, solicitante de reportes y consultas generales en el sistema.

Nombre Actor: Asistente Social

Perfil: Persona encargada de la OBP (Oficina de Beneficios a las Personas), solicitante de información respecto a las deudas o descuentos que puedan poseer los socios.

Nombre Actor: Administrador

Perfil: Persona responsable de mantener información de parámetros para el óptimo funcionamiento del sistema.

4.2 Diagramas de Casos de Uso

4.2.1 Caso de Uso General

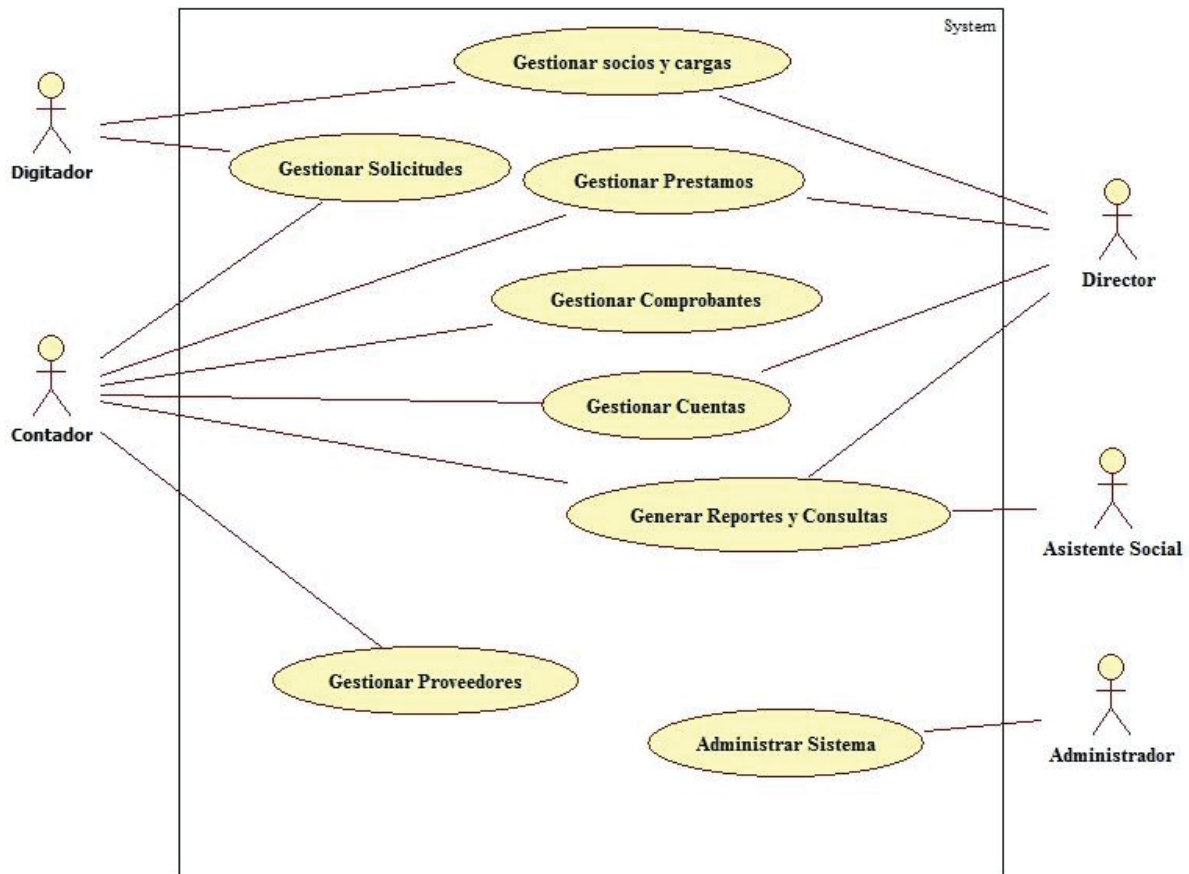


Figura 4.1 Caso de Uso General Primera Iteración

4.2.2 Casos de Uso Específicos

4.2.2.1 Caso de uso Gestionar Solicitudes

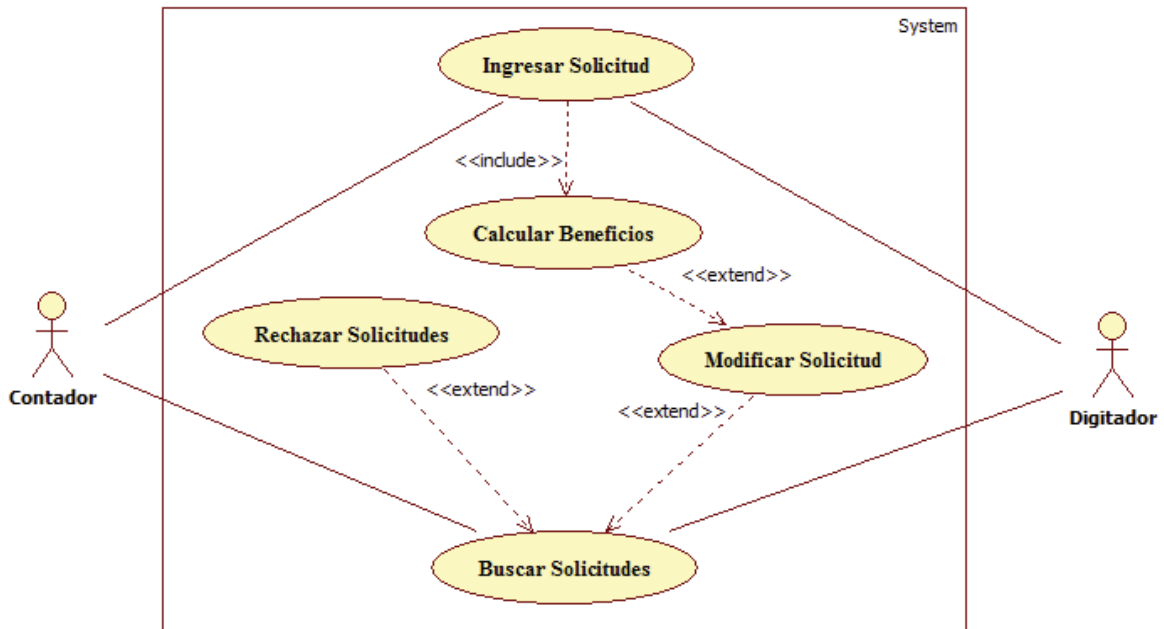


Figura 4.2 Gestionar Solicitudes (Ver Narrativo en Anexo A)

4.2.2.2 Caso de uso Gestionar Comprobantes

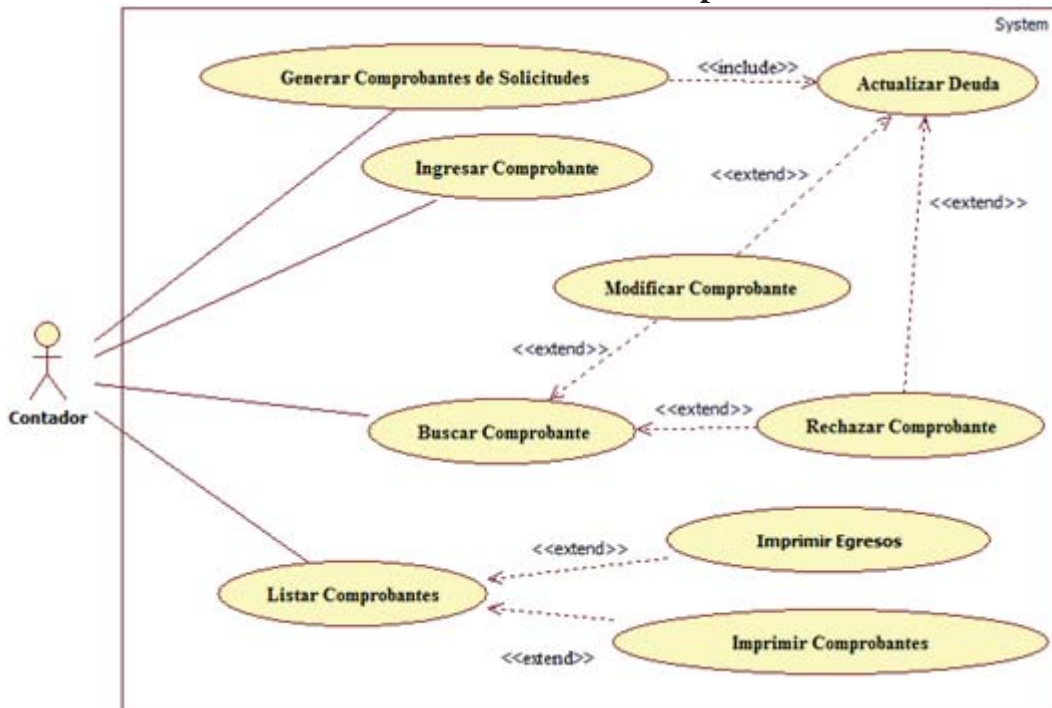


Figura 4.3 Gestionar Comprobantes (Ver Narrativo en Anexo B)

4.2.2.3 Caso de uso Generar Reportes y Contabilidad

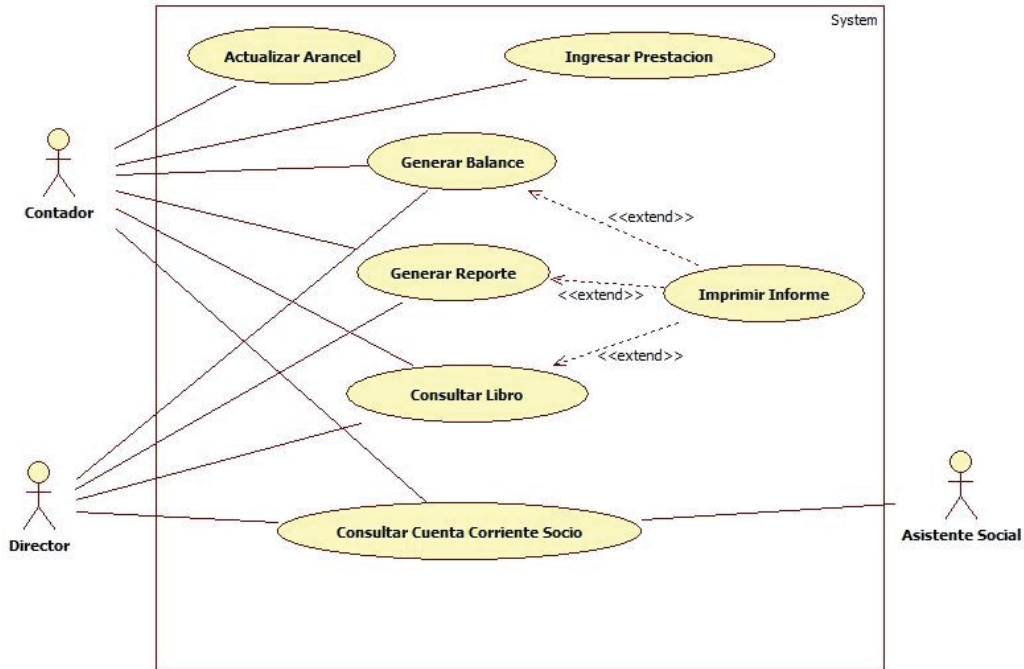


Figura 4.4 Generar Reportes y Contabilidad. (Ver Narrativo en Anexo C)

4.2.2.4 Caso de uso Gestionar Prestamos

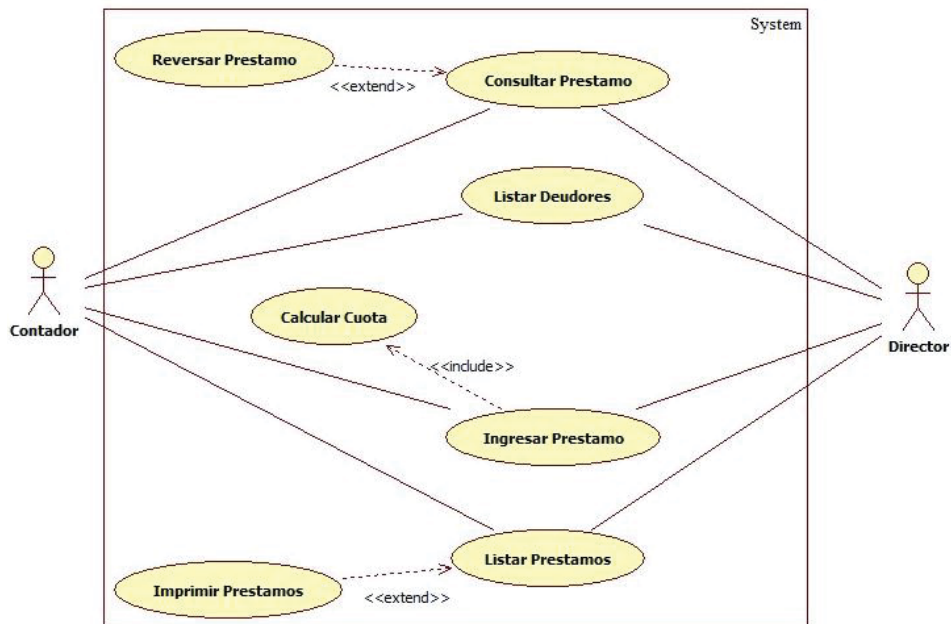


Figura 4.5 Gestionar Préstamos. (Ver Narrativo en Anexo D)

4.2.2.5 Caso de uso Gestionar Proveedores

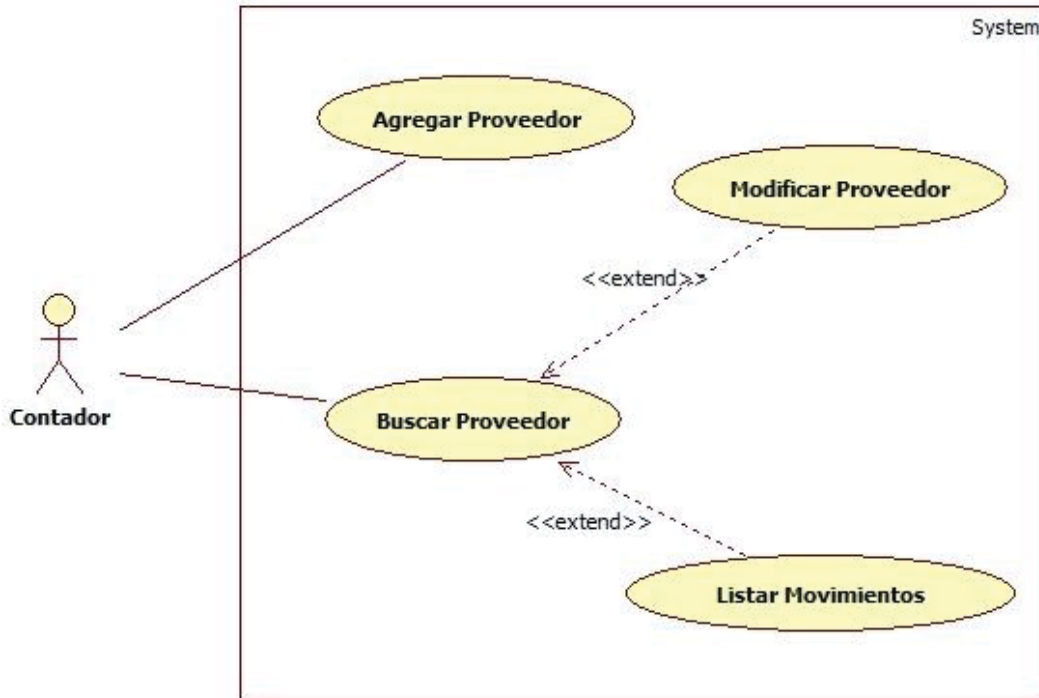


Figura 4.6 Gestionar Proveedores. (Ver Narrativo en Anexo E)

4.2.2.6 Caso de uso Gestionar Cuentas

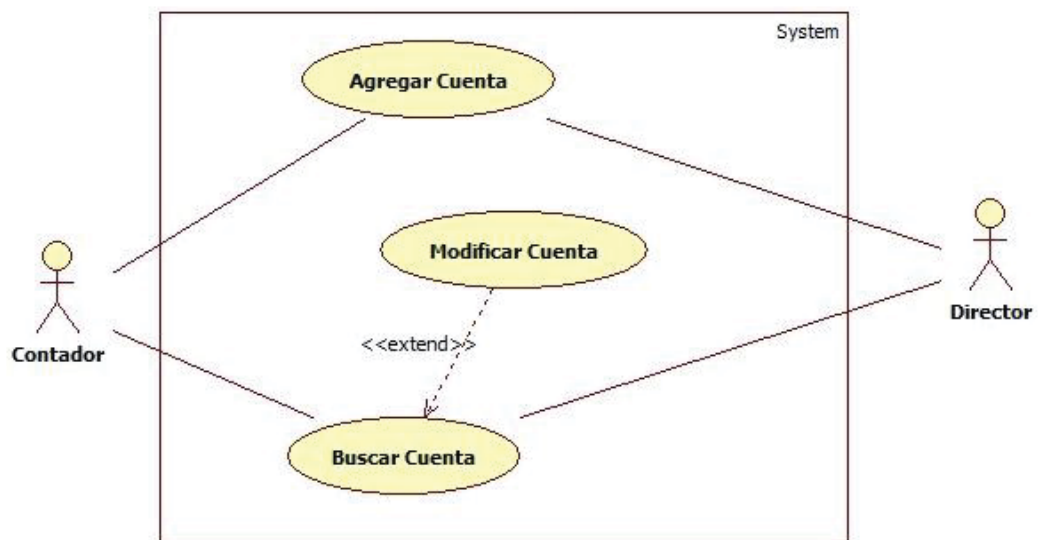


Figura 4.7 Gestionar Cuentas.

4.2.2.7 Caso de uso Administrar Usuarios

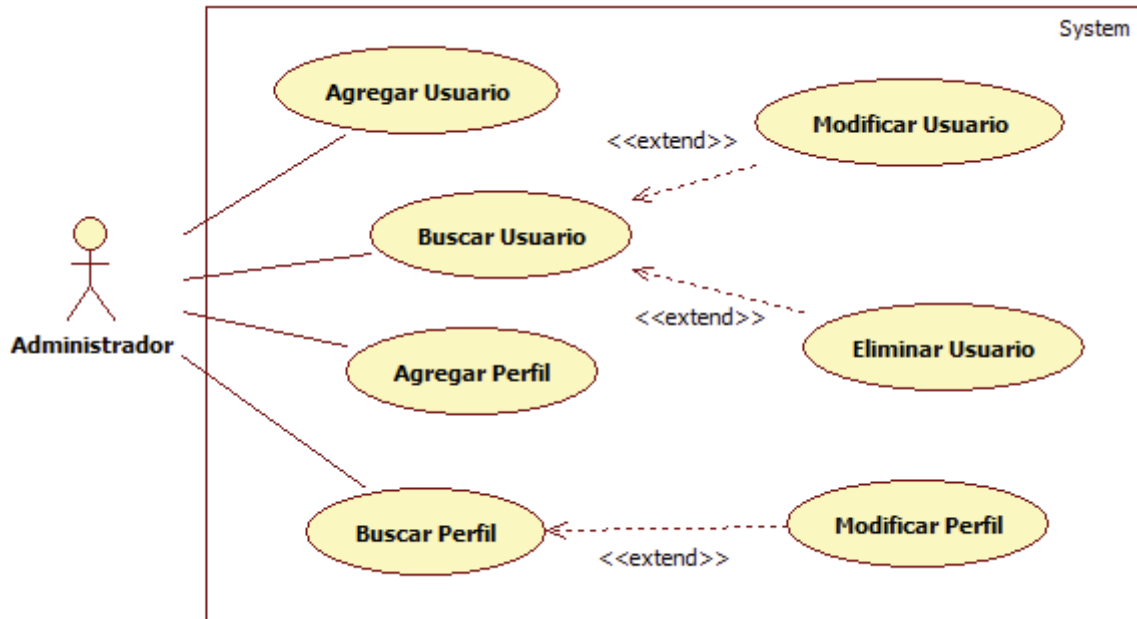


Figura 4.8 Administrar Usuarios.

4.2.2.8 Caso de uso Gestionar Socios y Cargas

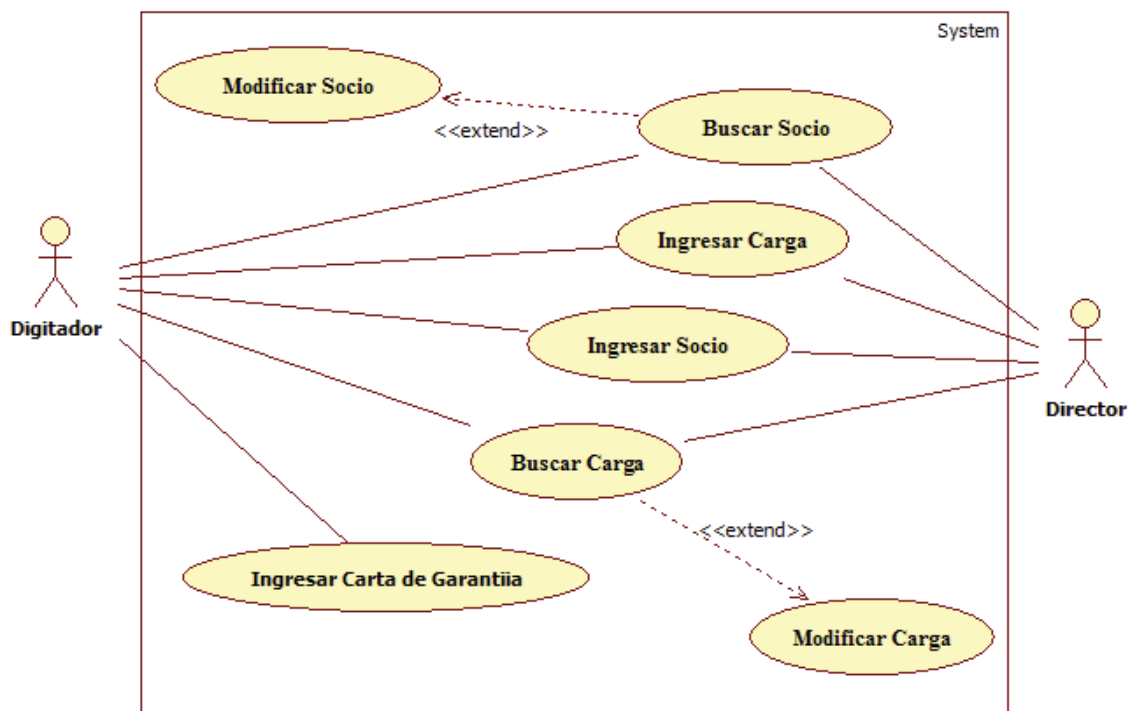


Figura 4.9 Gestionar Socios y Cargas.

5 Diseño Primera Iteración

5.1 Diagramas de Secuencia

Un diagrama de secuencia, corresponde a una representación que ilustra, en un determinado escenario, los eventos generados por actores externos, su orden y los eventos externos del sistema.

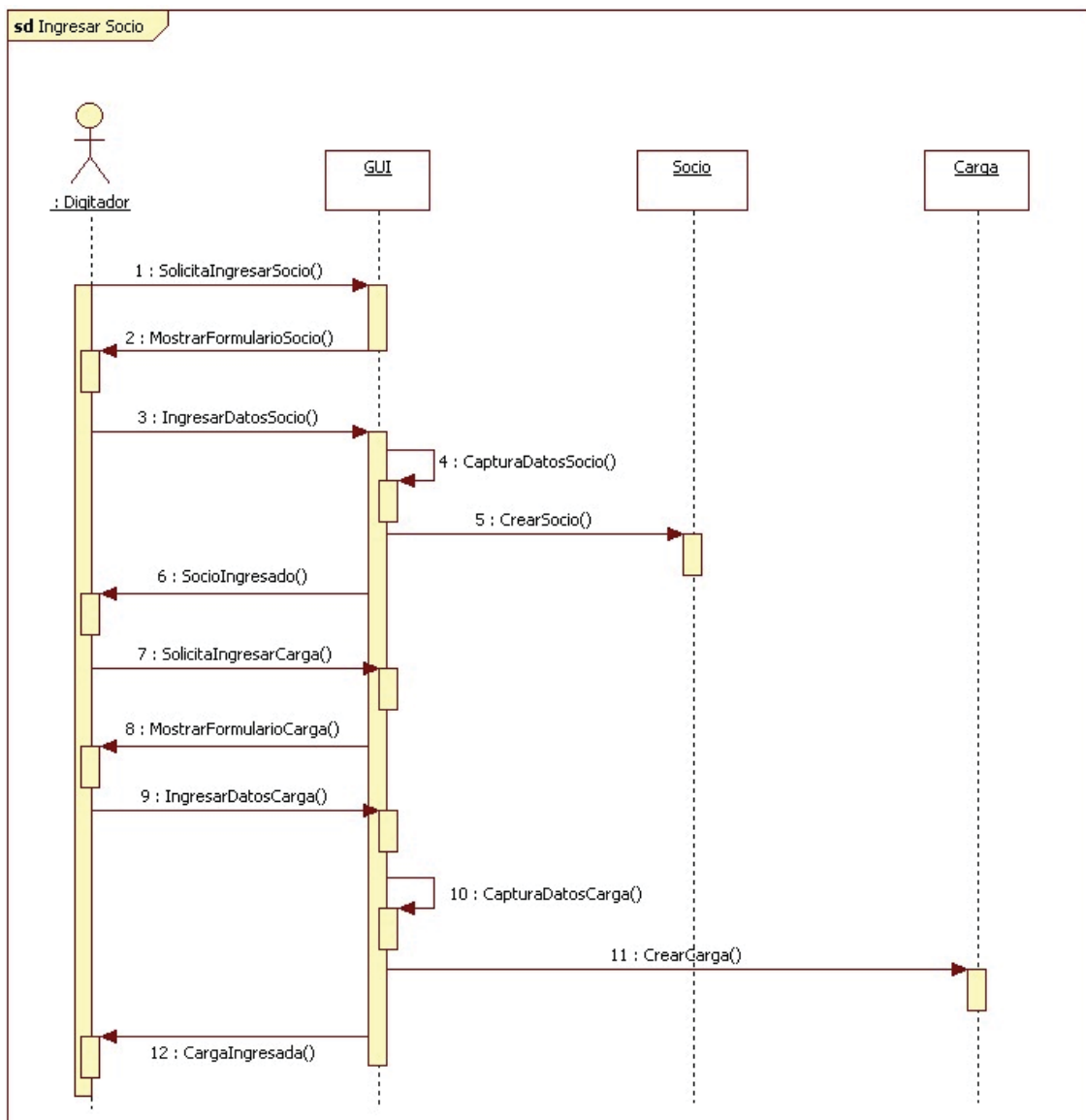


Figura 5.1 Diagrama de Secuencia Ingresar Socio

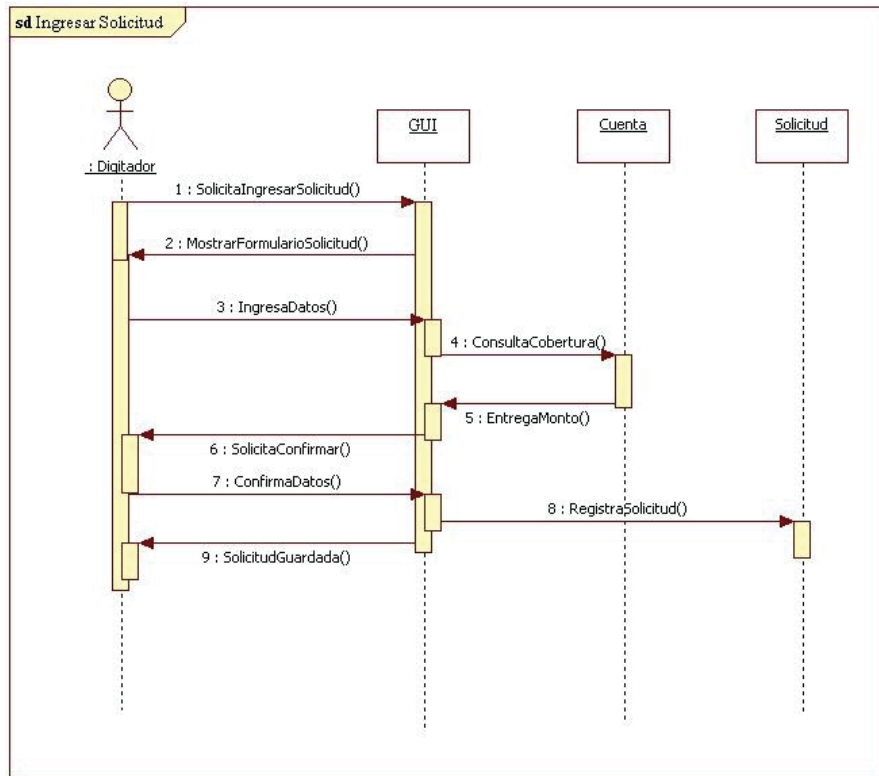


Figura 5.2 Diagrama de Secuencia Ingresar Solicitud

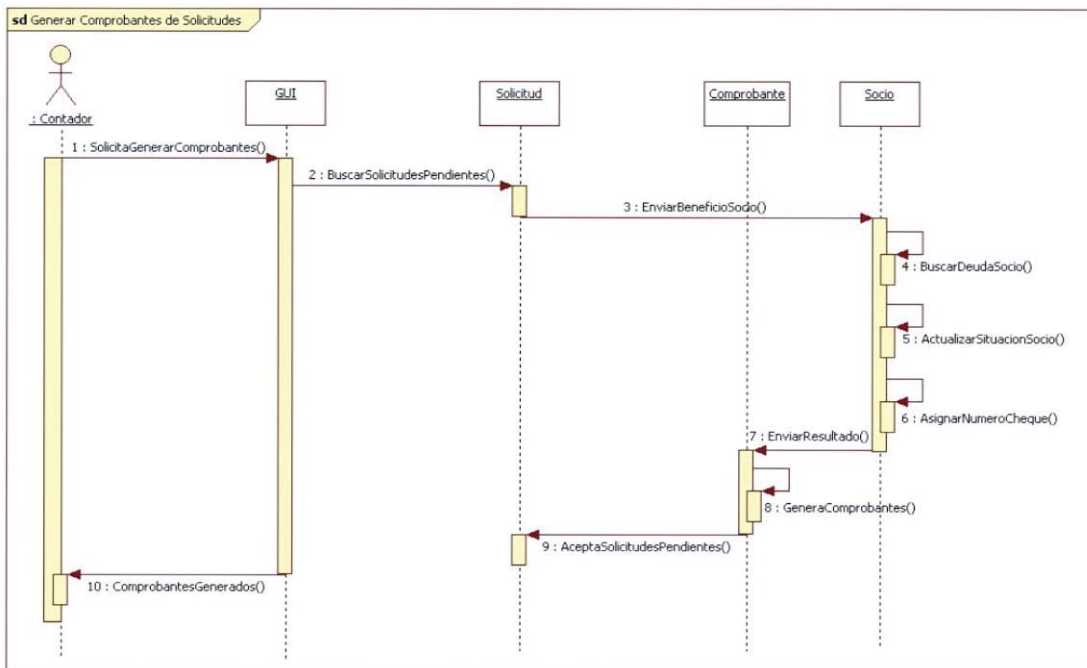


Figura 5.3 Diagrama de Secuencia Generar Comprobantes de Solicitudes.

5.2 Diagrama de Clase

Un diagrama de clases es un tipo de diagrama que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

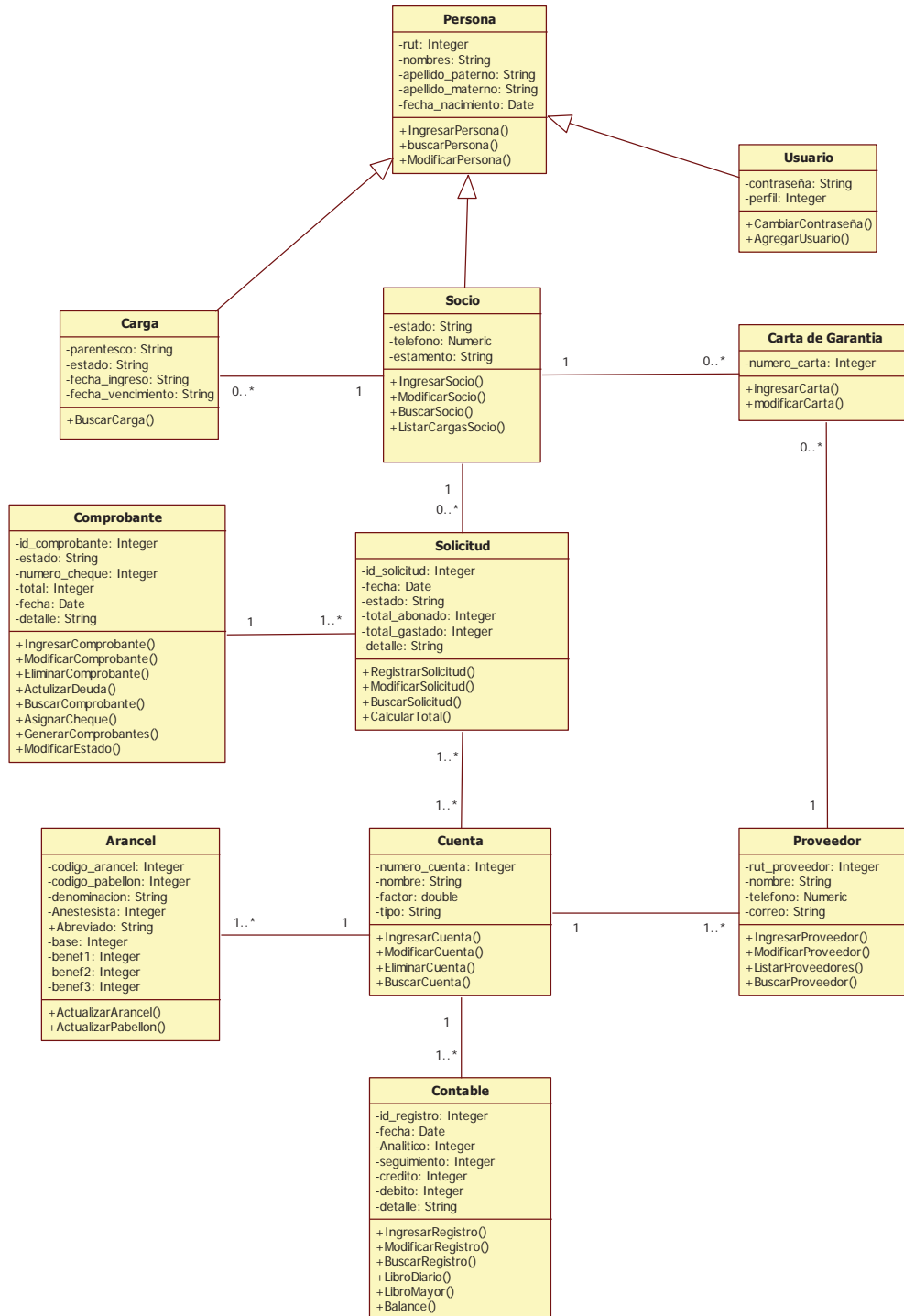


Figura 5.4 Diagrama de Clase Primera Iteración

6 Análisis Segunda Iteración

6.1 Requerimientos

6.1.1 Requerimientos Funcionales

Administración de Seguro Catastrófico

Manejo de eventos propios del seguro y las liquidaciones asociadas, tomando en cuenta el tiempo de vigencia de los eventos.

Mantenión de Empleados

Se generaliza el concepto de Socio para pasar a trabajar con todos los Empleados de la Universidad, incorporando sus datos personales así como su estado con el Bienestar y con el Seguro Catastrófico.

Manejo de Planilla para depósitos en Cuentas

Se incorpora la forma de pago de beneficios a través de depósitos en cuenta corriente o vista para los socios del bienestar que posean y en su defecto vale vista para los que no posean ningún tipo de cuenta.

Mantenión de Cargas

Incorporar las cargas legales que desee el empleado y manejar los vínculos que posean con el bienestar y su vigencia, así como también las cargas del seguro catastrófico las cuales no necesariamente son cargas legales.

Mantenión de Prestamos

Administración de préstamos otorgados no solamente por el Departamento de Bienestar, sino también por otros departamentos u oficinas, como lo es la Oficina de beneficios a las Personas (O.B.P.).

6.2 Diagramas de Casos de Uso

6.2.1 Caso de Uso General

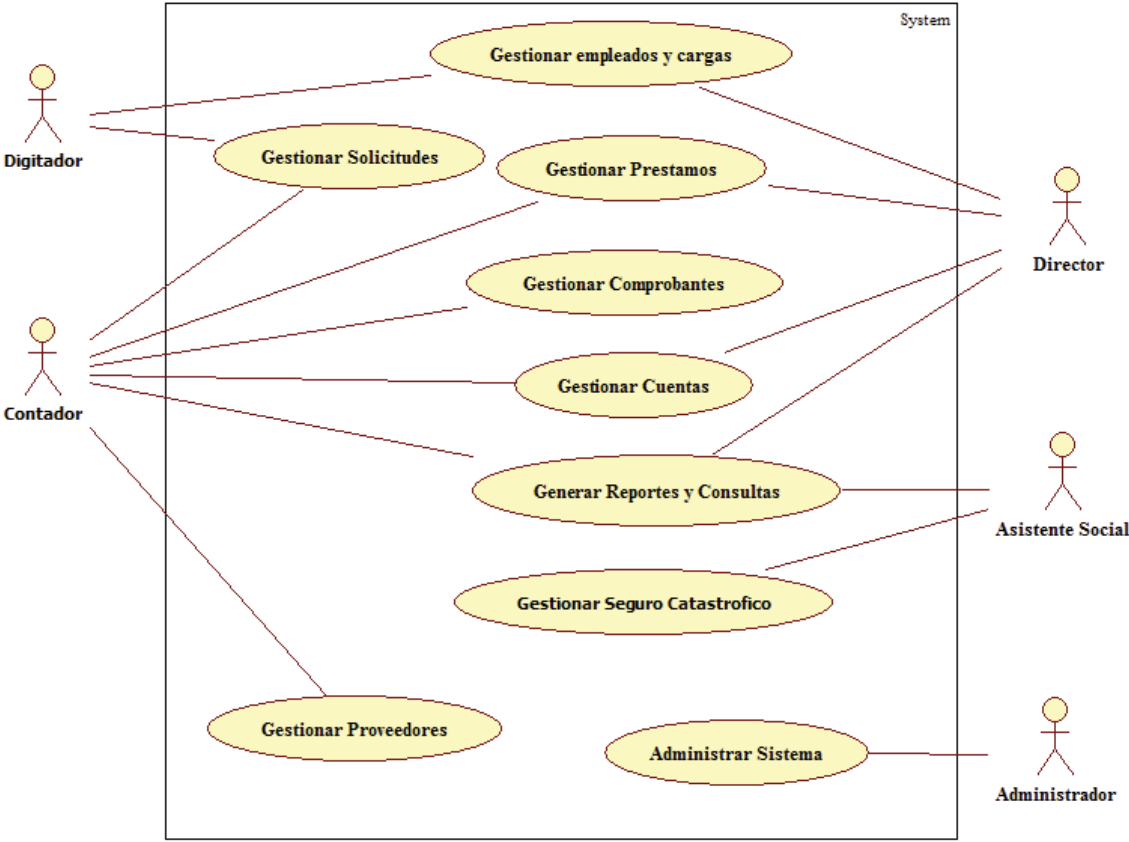


Figura 6.1 Caso de Uso General Segunda Iteración

6.2.2 Casos de Usos Específicos

6.2.1.1 Caso de Uso Gestionar Empleado

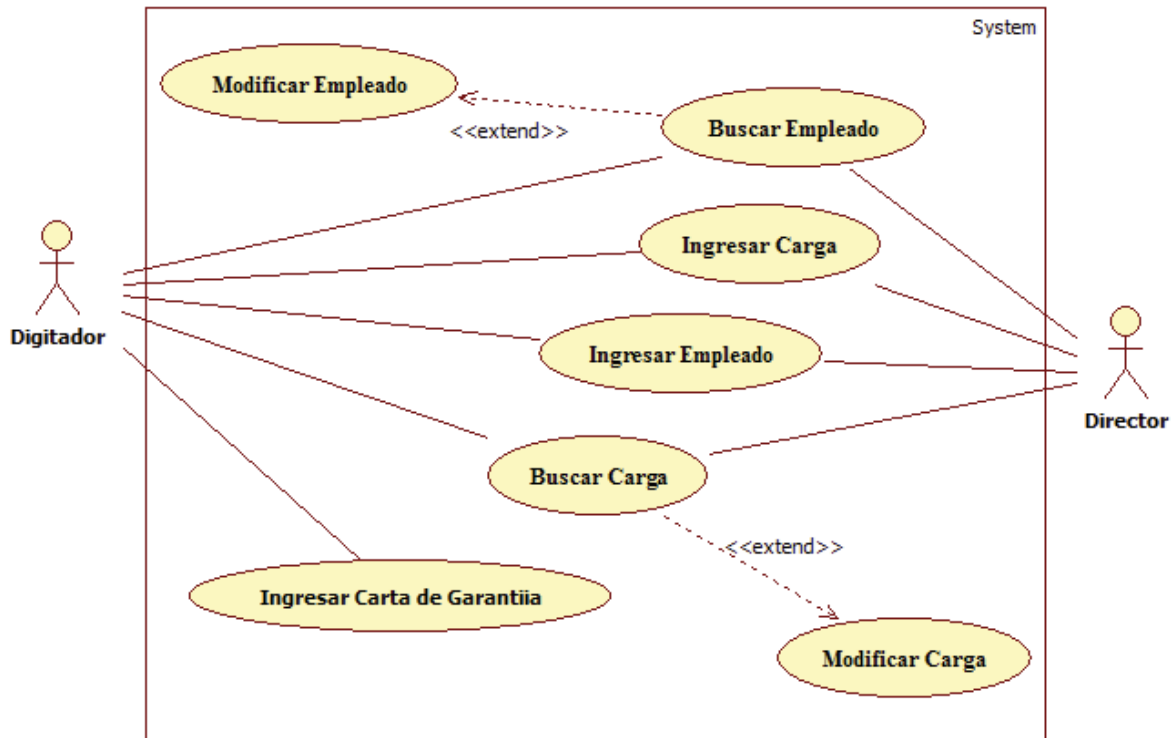


Figura 6.2 Caso de Uso Gestionar Empleado

6.2.1.2 Caso de Uso Gestionar Seguro Catastrófico.

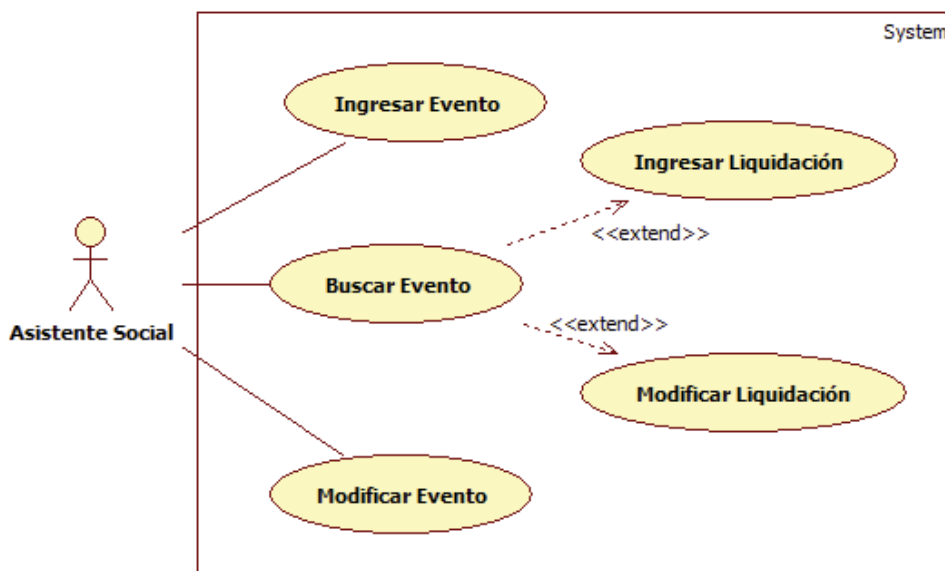


Figura 6.3 Caso de Uso Gestionar Seguro Catastrófico

7 Diseño Segunda Iteración

7.1 Diagrama de Clase

A continuación se presenta el diagrama de clase de la segunda iteración, obtenido de la iteración anterior, agregando los atributos y clases nuevas de los requerimientos.

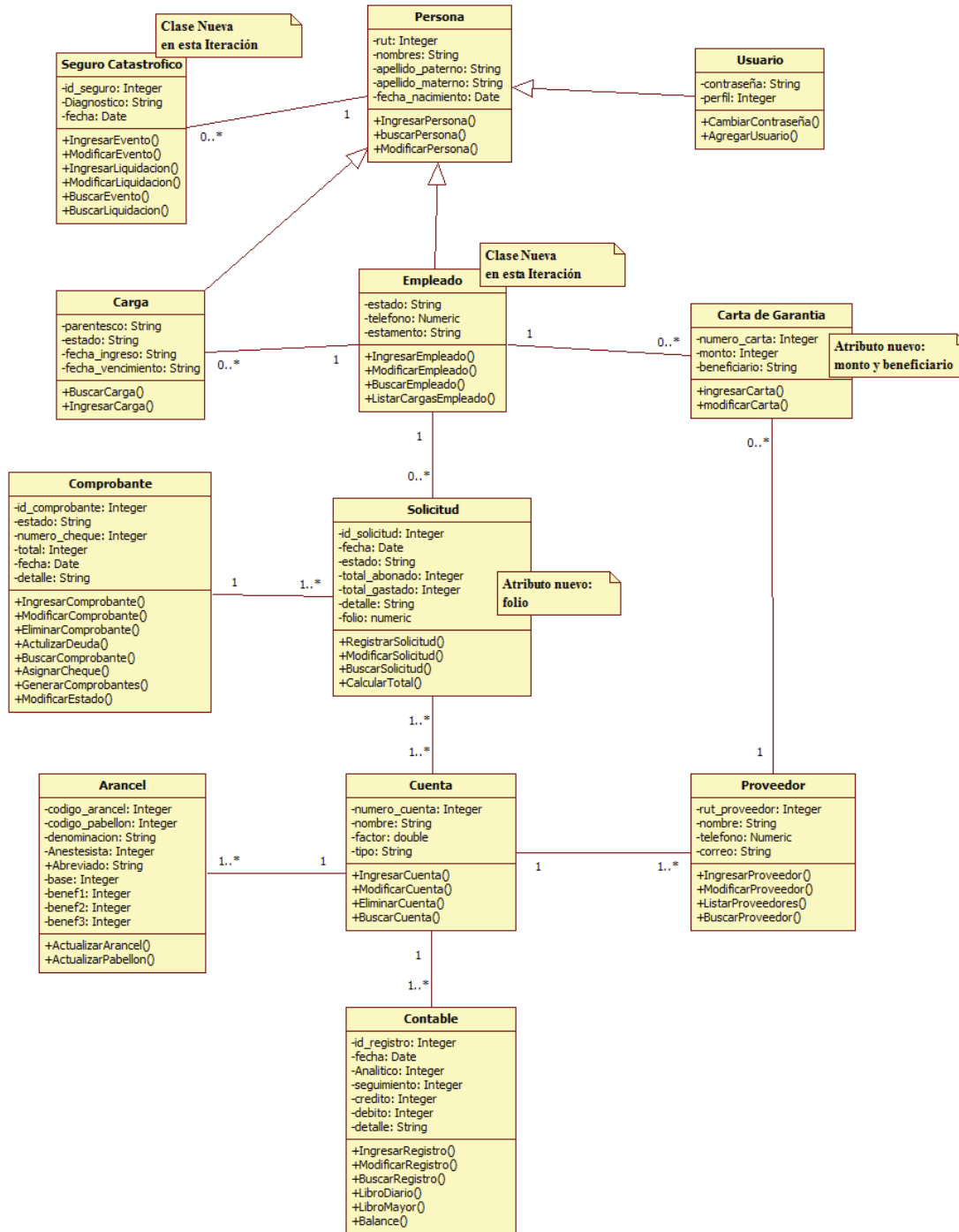


Figura 7.1 Diagrama de Clase Segunda Iteración

8 Arquitectura de la aplicación

La Arquitectura de Software se refiere a las estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos. La arquitectura de software es de especial importancia ya que la manera en que se estructura un sistema tiene un impacto directo sobre la capacidad de este para satisfacer lo que se conoce como los atributos de calidad del sistema. Ejemplos de atributos de calidad son el desempeño, que tiene que ver con el tiempo de respuesta del sistema a las peticiones que se le hacen, la usabilidad, que tiene que ver con qué tan sencillo les resulta a los usuarios realizar operaciones con el sistema.

El impacto también se puede apreciar sobre aspectos funcionales del sistema, ya que una elección errónea de la Arquitectura implicará errores a la hora de implementar el sistema. A continuación se describen las características de la arquitectura a utilizar, junto a las razones por la cual se optó por sobre otras.

8.1 Arquitectura Cliente-Servidor de tres capas

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Específicamente las tres capas se refieren a la distribución donde se ubica un nodo intermedio entre el cliente y el servidor encargado de procesar los datos para el cliente.

Se escogió la arquitectura Cliente-Servidor por las características de los requerimientos, específicamente la necesidad de ejecutar una operación más de un usuario a la vez y trabajar sobre la misma información, y se optó por las 3 capas debido a su capacidad de encapsular cada etapa haciendo más seguro el sistema, así como su mantención o posibles cambios a futuro.

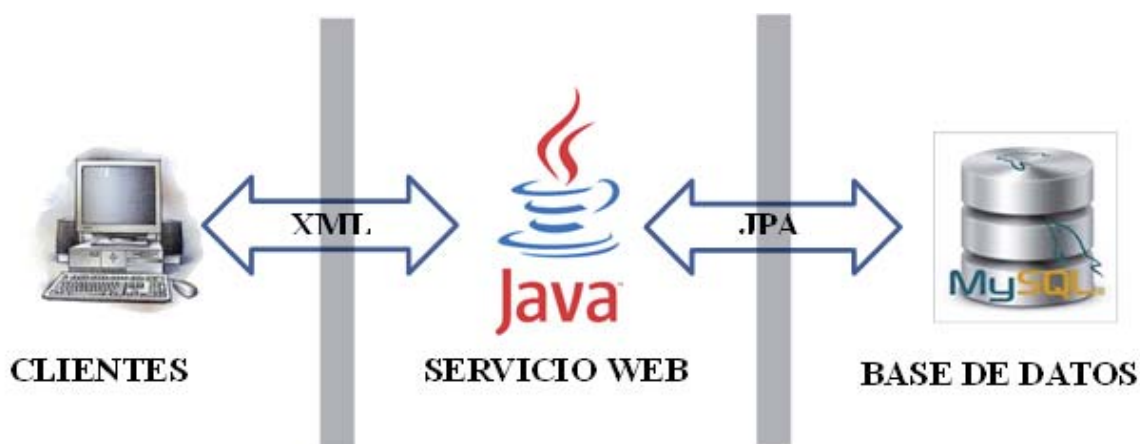


Figura 8.1 Arquitectura de la Aplicación

9 Implementación

Esta etapa se inició con la creación de un prototipo, el cual se desarrolló con los requerimientos más relevantes del proyecto.

En primera instancia se aplicó la arquitectura mencionada a través de la aplicación PowerDesigner creado el modelo físico de la base de datos, para luego ejecutar el script creado por la aplicación en el motor.

En una segunda etapa se procedió con la creación del servicio web en lenguaje JAVA, mediante la IDE Netbeans, dicho servicio se crea del lado del servidor y se conecta a la base de datos mediante un mapeo Objeto-relacional realizado por una API del lenguaje antes mencionado. En el servicio web se crea un código genérico destinado a ser usados por todos los métodos, el cual en caso de realizar una inserción, modificación o borrado de algún registro, se maneja mediante transacciones, en caso de consulta se retorna un String concatenando los datos necesarios.

```
@WebMethod(operationName = "NOMBRE_METODO")

public String NOMBRE_METODO(@WebParam(name = "nombreVariable") tipoDato
nombreVariable) {

Persistence.createEntityManagerFactory("bienestar_web04PU");

// llama a una conexión y permite utilizar las clases creadas a partir
del mapeo.

EntityManager em = emf.createEntityManager();

// crea una instancia de la base de datos

EntityTransaction et = em.getTransaction();// en caso de transacción

try{ //permite el manejo de errores

    et.begin();//inicia transacción

    .....

    et.commint(); // cierra transacción

    em.close();

    emf.close();

    return "creado";// palabra para identificar la operación exitosa

}catch(Exception ex){
```



```
// en caso de error ejecuta cierto código, evitando caídas de programa.  
    et.rollback(); // en caso de transacción  
    em.close(); // cierre instancia  
    emf.close(); // cierre de conexión  
    return ex.getMessage();  
// retorna un mensaje con el error correspondiente  
    }  
}
```

En la capa del lado del cliente, se crea un proyecto en la misma IDE que el servicio web, que trabaja con JAVA Swing, la cual es una librería destinada al manejo de ventanas, en interfaz de escritorio, este proyecto funciona como cliente del servicio web, destinado solo mostrar información para reducir los cálculos del lado de cliente, ejecutando la mayor cantidad de operaciones en el servicio web o en la base de datos.

Para el manejo de archivos se utiliza la librería JExcelAPI (jxl) para leer archivos de carga masiva en Excel, y en el caso de los archivos de salida necesarios en PDF, como lo es la contabilidad y los reportes se utiliza la librería iText. Dentro del proyecto del lado del cliente se utiliza una librería más para el manejo de interfaz gráfica de calendarios para la selección de fechas llamada JCalendar.

10 Plan de Pruebas

El plan de pruebas es el conjunto de actividades destinadas a realizar la verificación y validación de un atributo o capacidad de un programa o sistema para determinar que cumple con ciertas condiciones. La evaluación se estructura a partir de un enfoque en 2 condiciones, cumplir con todos los requerimientos y detectar la mayor cantidad de errores, defectos y fallas.

10.1 Tipo de Pruebas Utilizadas

Dentro del Proyecto se analizó los tipos de pruebas que demandaran un periodo de tiempo razonablemente acotado y a su vez que fueran confiable en su resultado a la hora de identificar los errores más comunes dentro del desarrollo de un proyecto.

Como es común a la hora de la implementación de un proyecto se realizan pruebas unitarias sobre todas las funcionalidades que contempla el proyecto de manera informal, para luego generar un documento formal especificando los detalles de cada una de las pruebas así como los parámetros evaluados y sus resultados.

El documento generado contempla un resumen de pruebas de caja negra utilizando la técnica de particionamiento de equivalencias, la cual se centra en identificar un conjunto de clases de pruebas representativas de grandes conjuntos de otras pruebas posibles, la idea es que el producto bajo prueba se comporte de la misma manera para todos los miembros de la clase.

Dicho documento consta de una tabla (ver Anexo H) con cada proceso del sistema, que se desglosa en todas las funcionalidades que involucran dichos procesos. Sobre estas funcionalidades se realizó un testeo en todos los campos de entrada, involucrando entradas de 3 formas distintas: dato válido dentro de rango aceptado, dato válido fuera de rango y dato no válido. Por ejemplo para la entrada Rut empleado se introdujo un número que estuviera en el sistema, para un dato válido fuera de rango se introdujo un número decimal y para un dato no válido se ingresó una cadena de caracteres. Para los datos probados se especificó una respuesta esperada por parte del sistema, llevándose a cabo una comparación con los resultados reales obtenidos desde el sistema.

11 Migración de Datos

Se denomina migración de datos, al proceso que tiene por objeto tanto la importación como la exportación de una determinada información almacenada en un sistema de bases de datos, para llevar a cabo su traspaso.

La migración de datos tiene su fundamento en la ampliación un sistema de gestión de base. En este contexto, se trata de exportar los datos a un nuevo sistema con mayor capacidad o más funciones adicionales. Estos cambios llevan consigo una adaptación de todos los datos de una base de datos a otra. Por tanto siempre que se producen cambios de un sistema de gestión a otro, se habla inevitablemente de los procesos de migración de datos.

11.1 Método a Utilizar para la Carga Inicial

Como punto de partida en cualquier migración se debe tener claro la estructura de origen y destino de los datos, para este proyecto los datos históricos del sistema actual se encuentran almacenados en formato dBase (.dbf), el cual posee la misma estructura tanto en el origen como en la tabla de destino, por lo que el traspaso de información contempla un proceso que consta de 3 etapas por cada registro.

En primera instancia se asignara un identificador único por registro, esto debido a que la identificación que se asigna en el sistema actual se reinicia cada cierto periodo de tiempo dando como resultado múltiples registros con el mismo identificador; en segunda etapa se realiza una conversión a los tipos de datos que se asignara en el destino ya que todas las columnas esta almacenadas de la misma manera; y finalmente se almacena el registro en la base de datos, todo esto mediante un proceso iterativo que leerá el archivo de origen desde el lado del cliente, pasando por el servicio web para realizar las inserciones respectivas.

12 Conclusión

El desarrollo de un proyecto informático, puede enfocarse de distintas formas, y se puede manifestar divergencia a la hora de analizar la manera como se desarrolla, sin embargo, la aplicación de metodologías formales, objetivos claros y bien definidos, y una planificación acertada y realista conducen a la obtención de un producto final de calidad.

En la Pontificia Universidad Católica de Valparaíso así como en cualquier organización de amplio ámbito, la planificación está organizada en función de sus objetivos, por lo que debe estar sujeta a un proceso formal, lo cual no quiere decir que sea rígida, por el contrario esta debe ser flexible y cambiante, ya que dentro de una organización los objetivos pueden sufrir cambio. También se hace importante la revisión de esta periódicamente y en los casos que se requiera la modificación, ya que de acuerdo a los resultados obtenidos, se podrá ejecutar una planificación cada vez más cercana a la realidad, haciendo futuras estimaciones más precisas.

Es importante señalar que dentro de cualquier proyecto que incorpore una organización donde ya se cuenta con un sistema informático, debe considerar un aspecto relevante como es la posible resistencia al cambio de parte de los usuarios, sobre todo los que interactúan en mayor medida con el sistema, esto se puede apreciar fuertemente en este proyecto, ya que el sistema con el que se cuenta actualmente cuenta con un funcionamiento satisfactorio de acuerdo a las necesidades anteriores, y un uso de alrededor de 25 años de parte de los mismos usuarios. Lo que nos lleva a concluir que cualquier posible solución informática dentro del Bienestar debe considerar minimizar el impacto dentro de su estructura de trabajo, esto se puede lograr de muchas maneras, una de ellas es involucrándolos en la etapa de desarrollo, haciéndolos participe de este cambio, tomando en cuenta sus opiniones.

Para finalizar es relevante mencionar que el departamento de Bienestar se encuentra en una etapa de transición, donde los usuarios finales de este sistema coexisten y se involucran con las etapas finales del desarrollo para obtener buenos resultados en las pruebas de adaptación, por lo que se espera que llegue a la etapa final el sistema desarrollado en este proyecto y se logre integrar al departamento. Como trabajo a futuro, se tiene una visión del amplio uso que se puede otorgar al sistema en el ámbito de préstamos dentro de la organización, debido a que la estructura permite el trabajo de esta funcionalidad independiente del departamento que lo lleve a cabo.

13 Referencias

[1] SCHMULLER: Aprendiendo UML en 24 Horas.

[2] PRESSMAN R.: Ingeniería de Software: Un Enfoque Práctico, Editorial McGraw-Hill, Quinta Edición, Año 2002.

[3] Deitel, Paul J. Deitel, Harvey M.: Java: Cómo Programar, Editorial Pearson Educación, Séptima Edición, Año 2008.

14 Anexos

A: Caso de Uso Narrativo Gestionar Solicitudes

Tabla 9.1 Caso de uso Ingresar Solicitud.

Caso de uso	Ingresar Solicitud
Actores	Contador, Digitador
Propósito	Ingresar al sistema una solicitud de reembolso.
Precondición	Debe estar identificado el usuario.
Post condición	
Escenario principal	<ol style="list-style-type: none"> 1. El Actor selecciona Ingresar Solicitud 2. El sistema despliega un formulario para ser completado por el actor. 3. El Actor ingresa todos los datos, tanto del socio como de la(s) prestación(es) las cuales se desea reembolsar. 4. El Sistema calcula los beneficios asociados a los gastos incurridos por el socio. 5. El Actor confirma que los datos están correctos. 6. El Sistema almacena la solicitud que se desea procesar.
Escenario alternativo	
Puntos de extensión	
Observaciones	Incluye el caso de uso Calcular Beneficios.

Tabla 9.2 Caso de uso Calcular Beneficios.

Caso de uso	Calcular Beneficios
Actores	Contador, Digitador.
Propósito	Calcular los beneficios asociados a cada prestación que pueda tener una solicitud.
Precondición	Debe estar identificado el usuario además debe encontrarse en la opción Ingresar Solicitud.
Post condición	

Escenario principal	<ol style="list-style-type: none"> 1. El Actor confirma que desea guardar la solicitud ingresada. 2. El sistema previo al almacenado de la solicitud calcula los beneficios propios de la(s) prestación(es) que componen la antes mencionada.
Escenario alternativo	
Puntos de extensión	
Observaciones	

B: Caso de Uso Narrativo Gestionar Comprobantes

Tabla 9.3 Caso de uso Generar Comprobantes de Solicitudes.

Caso de uso	Generar Comprobantes de Solicitudes
Actores	Contador
Propósito	Generar los comprobantes, ya sea abono a deuda o pago a socio, asociados a las solicitudes pendientes.
Precondición	Debe estar identificado el usuario, debe haber al menos una solicitud pendiente y la(s) solicitud(es) pendientes deben encontrarse ingresadas de forma correcta.
Post condición	
Escenario principal	<ol style="list-style-type: none"> 1. El Actor ingresa en la opción generar comprobantes de solicitudes. 2. El sistema muestra la(s) solicitud(es) pendiente(s). 3. El Actor selecciona generar comprobantes. 4. El sistema previa validación de la situación del socio (Actualizar Deuda) guarda los comprobantes generados.
Escenario alternativo	
Puntos de extensión	
Observaciones	Incluye el caso de uso “Actualizar Deuda”.

Tabla 9.4 Caso de uso Actualizar Deuda.

Caso de uso	Actualizar Deuda
Actores	Contador
Propósito	Mantener la deuda que pueda tener el socio al día.
Precondición	Debe estar identificado el usuario.
Post condición	
Escenario principal	<ol style="list-style-type: none"> 1. El Actor confirma que desea generar los comprobantes de solicitudes. 2. El sistema verifica por cada socio su deuda, en caso de poseer una deuda genera un comprobante para abono a su deuda en caso contrario un comprobante de tipo 2 (Egreso).
Escenario alternativo	
Puntos de extensión	
Observaciones	

Tabla 9.5 Caso de uso Buscar Comprobante.

Caso de uso	Buscar Comprobante
Actores	Contador
Propósito	Poder realizar una consulta de cierto comprobante ya sea para identificar o modificar este.
Precondición	Debe estar identificado el usuario, debe existir el número del comprobante que se desea buscar.
Post condición	
Escenario principal	<ol style="list-style-type: none"> 1. El Actor ingresa en la opción buscar comprobante. 2. El sistema muestra la información asociada al comprobante buscado. 3. El sistema muestra las operaciones posibles sobre el comprobante.
Escenario alternativo	
Puntos de extensión	-Se Extiende al caso de uso Modificar Comprobante o Rechazar Comprobante depende de cuál sea el caso.

Tabla 9.6 Caso de uso Ingresar Comprobante.

Caso de uso	Ingresar Comprobante
Actores	Contador
Propósito	Ingresar un comprobante que no provenga de una solicitud
Precondición	Debe estar identificado el usuario.
Post condición	
Escenario principal	<ol style="list-style-type: none"> 1. El Actor selecciona la opción ingresar comprobante 2. El sistema muestra el formulario con los datos mínimos para que se almacene el comprobante. 3. El Actor completa el formulario y lo envía. 4. El sistema actualiza las cuentas contables respectivas.
Escenario alternativo	
Puntos de extensión	
Observaciones	Incluye el caso de uso “Actualizar Deuda”.

C: Caso de Uso Narrativo Generar Reportes y Consultas

Tabla 9.7 Caso de uso Consultar Movimientos Socio.

Caso de uso	Consultar Movimientos Socio
Actores	Contador, Asistente Social y Director
Propósito	Consultar movimiento por socio para saber estado de sus descuentos aplicados a la liquidación, así como los beneficios cobrados.
Precondición	Debe estar identificado el usuario, debe existir al menos un movimiento para el socio indicado.
Post condición	
Escenario principal	<ol style="list-style-type: none"> 1. El Actor selecciona la opción “Consultar Movimiento por Socio”. 2. El sistema muestra un formulario que debe completar el actor. 3. El Actor ingresa los datos del socio así como los tipos de cuentas que se desean ver.

Escenario alternativo	4. El sistema muestra tanto las deudas como los beneficios cobrados por el socio.
Puntos de extensión	
Observaciones	

Tabla 9.8 Caso de uso Actualizar Arancel.

Caso de uso	Actualizar Arancel
Actores	Contador
Propósito	Realizar de manera automática la actualización de los montos del arancel Fonasa.
Precondición	Debe estar identificado el usuario, deben estar ingresadas todas las prestaciones ya que es solo una actualización de los montos.
Post condición	
Escenario principal	<ol style="list-style-type: none"> 1. El Actor selecciona la opción “Actualizar Arancel”. 2. El sistema solicita seleccionar la ubicación del archivo donde están los montos nuevos. 3. El Actor ingresa la ubicación del archivo. 4. El sistema actualiza los montos de cada prestación médica.
Escenario alternativo	4.1. En caso que exista(n) un código nuevo, el sistema dará la opción de agregarlo de forma manual.
Puntos de extensión	
Observaciones	

Tabla 9.9 Caso de uso Consultar Libro.

Caso de uso	Consultar Libros
Actores	Contador, Director
Propósito	Realizar consultas sobre movimientos específicos en el libro Diario o en el libro Mayor, ya sea del día en si, como de otro periodo en particular.
Precondición	Debe estar identificado el usuario.
Post condición	

Escenario principal	<ol style="list-style-type: none"> 1. El Actor selecciona la opción “Consultar Libro”. 2. El sistema muestra un formulario que debe completar el actor para desplegar la información necesaria. 3. El Actor ingresa los datos del movimiento(s) que se desea(n) buscar.
Escenario alternativo	
Puntos de extensión	
Observaciones	

Tabla 9.10 Caso de uso Generar Balance.

Caso de uso	Generar Balance
Actores	Contador, Director.
Propósito	Generar el balance mensual analizando todas las cuentas contables del sistema.
Precondición	Debe estar identificado el usuario.
Post condición	
Escenario principal	<ol style="list-style-type: none"> 1. El Actor selecciona la opción “Generar Balance”. 2. El sistema muestra la información solicitada. 3. El sistema despliega la opción de imprimir el balance generado.
Escenario alternativo	3.1. Se desplegara la opción de imprimir el balance en caso que el usuario lo requiera de lo contrario por defecto se guardara en la base de datos.
Puntos de extensión	Se extiende el caso de uso “Imprimir Informe”.
.Observaciones	

D: Caso de Uso Narrativo Gestionar Prestamos

Tabla 9.11 Caso de uso Listar Deudores.

Caso de uso	Listar Deudores
Actores	Contador, Director.

Propósito	Obtener de forma actualizada todos los socios que no se les ha mandado a descuento su deuda.
Precondición	Debe estar identificado el usuario, debe existir al menos un socio con deuda.
Post condición	
Escenario principal	1. El Actor selecciona la opción “Listar Deudores”. 2. El sistema muestra los datos del socio así como la deuda que posee.
Escenario alternativo	
Puntos de extensión	Se extiende el caso de uso “Ingresar Préstamo”.
.Observaciones	

Tabla 9.12 Caso de uso Ingresar Préstamo.

Caso de uso	Ingresar Préstamo
Actores	Contador, Director.
Propósito	Enviar a descuento la deuda que posee el socio actualizando las cuentas contables asociadas de forma automática.
Precondición	Debe estar identificado el usuario, el socio debe poseer deuda.
Post condición	
Escenario principal	1. El Actor selecciona la opción “Ingresar Préstamo”. 2. El sistema muestra la deuda junto con un formulario para ingresar cuotas, mes y año en que se inician los descuentos. 3. El Actor ingresa los datos requeridos por el sistema para llevar la deuda a préstamo. 4. El sistema calcula el monto de las cuotas aplicando el interés respectivo, lo descuenta de y almacena la información, actualizando las cuentas contables respectivas.
Escenario alternativo	4.1 En caso de ser el número de cuotas igual o inferior a 3, se calcula el monto de la cuota sin interés.
Puntos de extensión	
.Observaciones	Incluye el caso de uso “Calcular Cuota”, donde se hace el cálculo respecto del interés y las amortizaciones que se pagaran mensualmente.

Tabla 9.13 Caso de uso Imprimir Préstamos.

Caso de uso	Imprimir Préstamos
Actores	Contador, Director.
Propósito	Imprimir el listado de los socios a los cuales se les debe aplicar descuento en el mes actual.
Precondición	Debe estar identificado el usuario.
Post condición	
Escenario principal	<ol style="list-style-type: none"> 1. El Actor selecciona la opción “Imprimir Préstamos”. 2. El sistema muestra una vista previa del listado de los socios que tiene que cancelar una cuota en el mes actual. 3. El Actor confirma los datos y selecciona imprimir.
Escenario alternativo	
Puntos de extensión	

E: Caso de uso Gestionar Proveedores

Tabla 9.14 Caso de uso Agregar Proveedor.

Caso de uso	Agregar Proveedor.
Actores	Contador.
Propósito	Incorporar al sistema un proveedor junto con sus datos para manejar de mejor manera estadísticas y control de los movimientos con ellos.
Precondición	Debe estar identificado el usuario.
Post condición	
Escenario principal	<ol style="list-style-type: none"> 1. El Actor selecciona la opción “Agregar Proveedor”. 2. El sistema despliega un formulario que debe llenar el usuario para ingresar el proveedor al sistema. 3. El Actor completa el formulario y confirma que la información es verídica. 4. El sistema confirma que la información se ha almacenado con éxito.

Escenario alternativo	
Puntos de extensión	
.Observaciones	Los casos de uso “Agregar Cuenta” y “Agregar Usuario” poseen la misma secuencia por lo que no se detallara, en sus respectivos casos de uso específico.

Tabla 9.15 Caso de uso Buscar Proveedor.

Caso de uso	Buscar Proveedor.
Actores	Contador.
Propósito	Buscar un proveedor por Rut o por nombre, para consultar algún dato de él.
Precondición	Debe estar identificado el usuario, debe existir al menos un proveedor.
Escenario principal	<ol style="list-style-type: none"> 1. El Actor selecciona la opción “Buscar Proveedor”. 2. El sistema muestra un formulario con los datos que se requieren para filtrar la búsqueda. 3. El Actor ingresa el(los) dato(s) del proveedor. 4. El sistema muestra la información del proveedor.
Escenario alternativo	
Puntos de extensión	Se extiende los casos de uso “Modificar Proveedor” y “Eliminar Proveedor”.
.Observaciones	Los casos de uso “Buscar Cuenta” y “Buscar Usuario” poseen la misma secuencia por lo que no se detallara, en sus respectivos casos de uso específico.

F: Modelo de Datos Primera Iteración

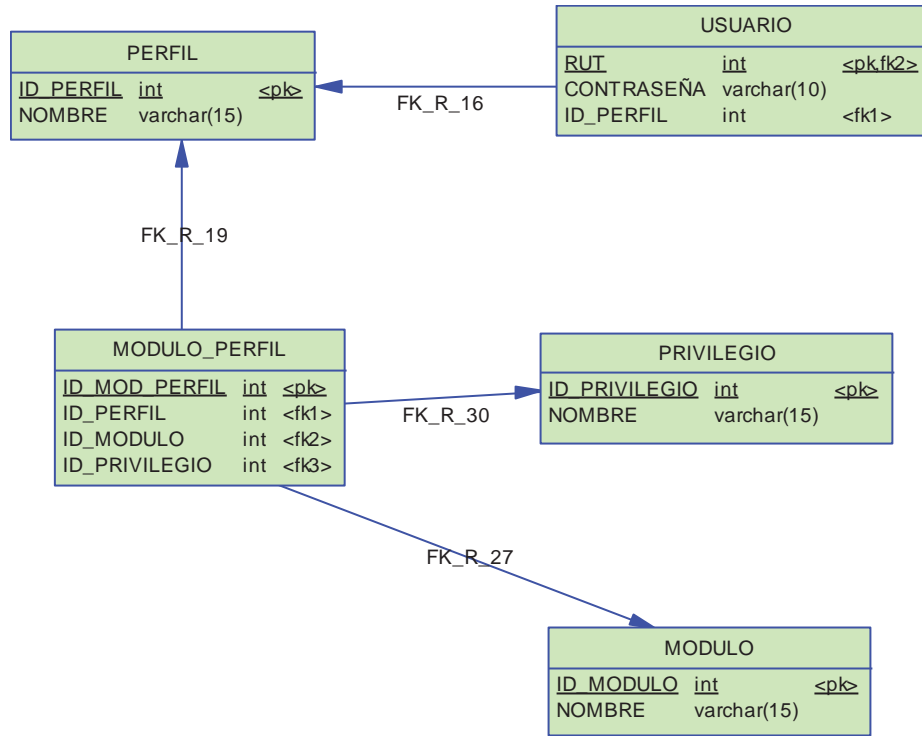


Figura 9.1: Sección relativa a USUARIOS

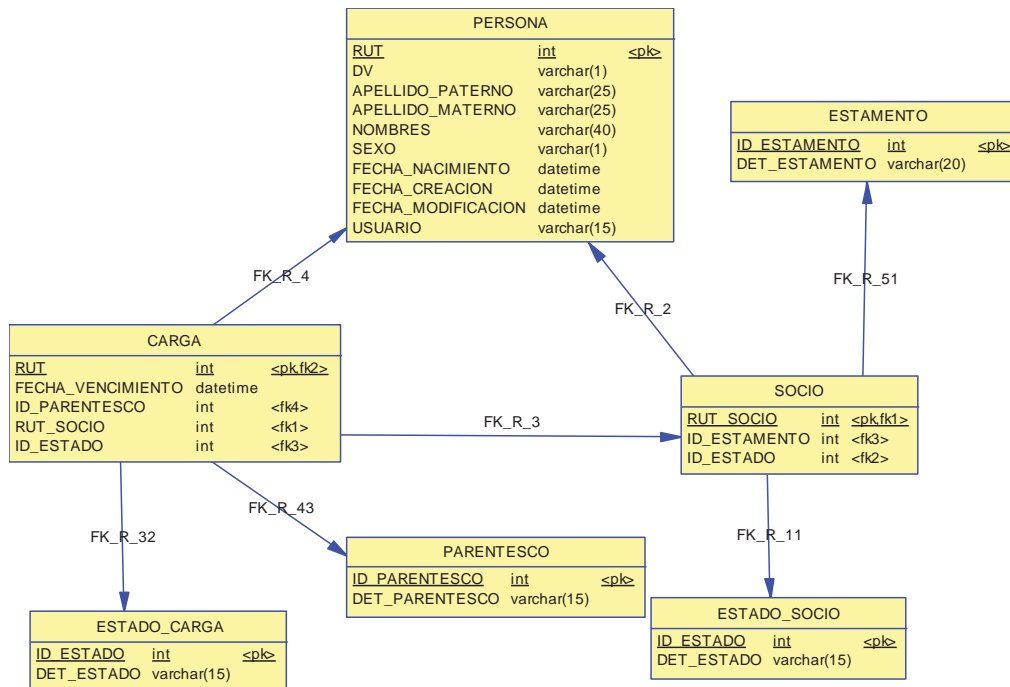


Figura 9.2: Sección relativa a PERSONA

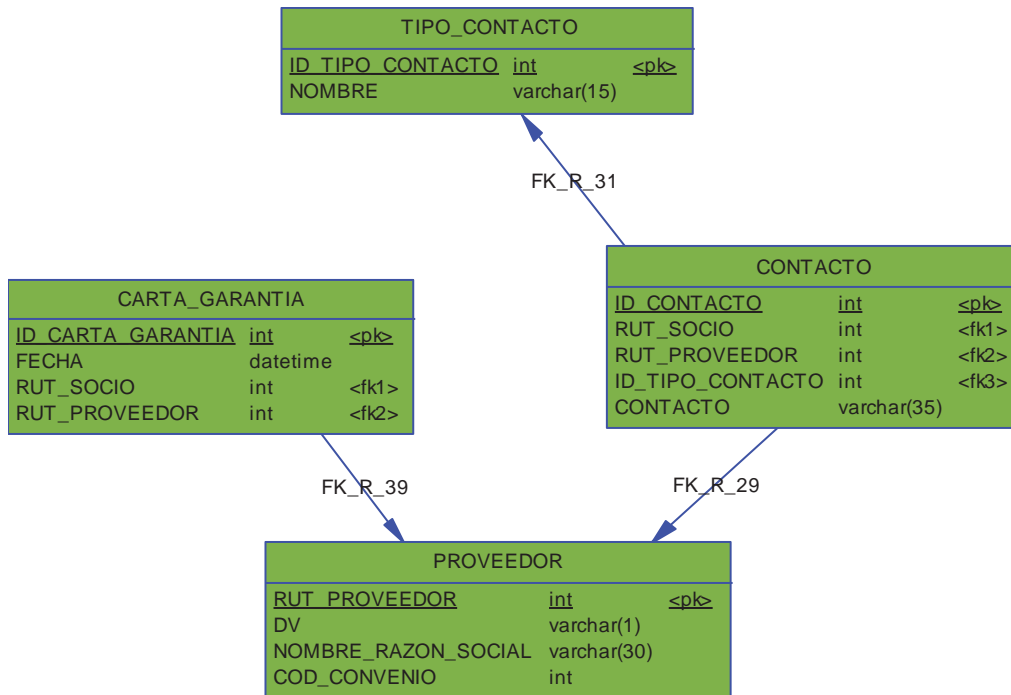


Figura 9.3: Sección relativa a PROVEEDOR

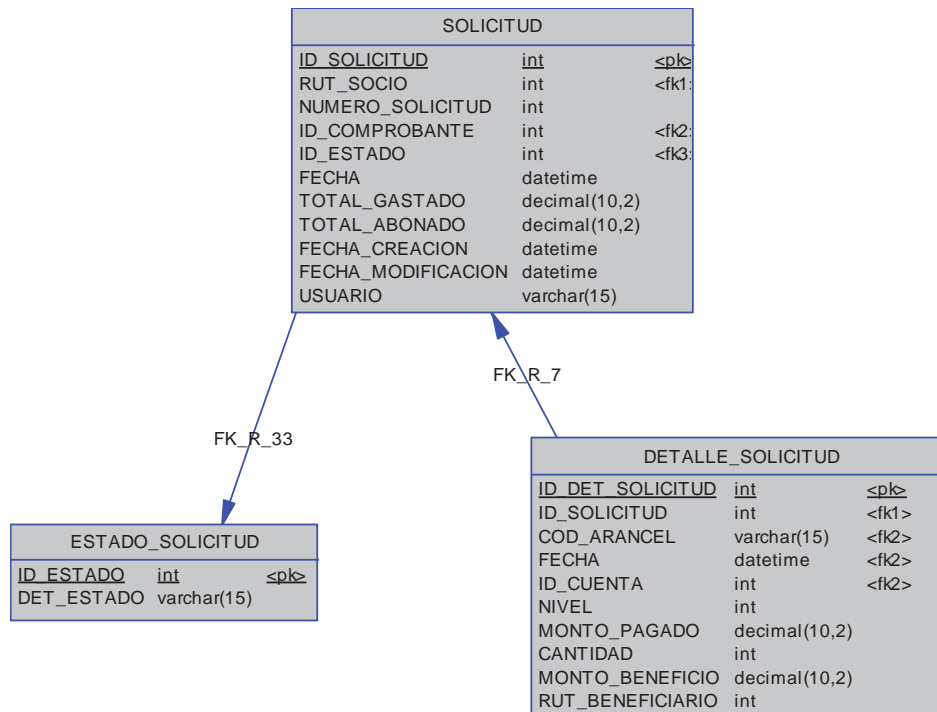


Figura 9.4: Sección relativa a SOLICITUD

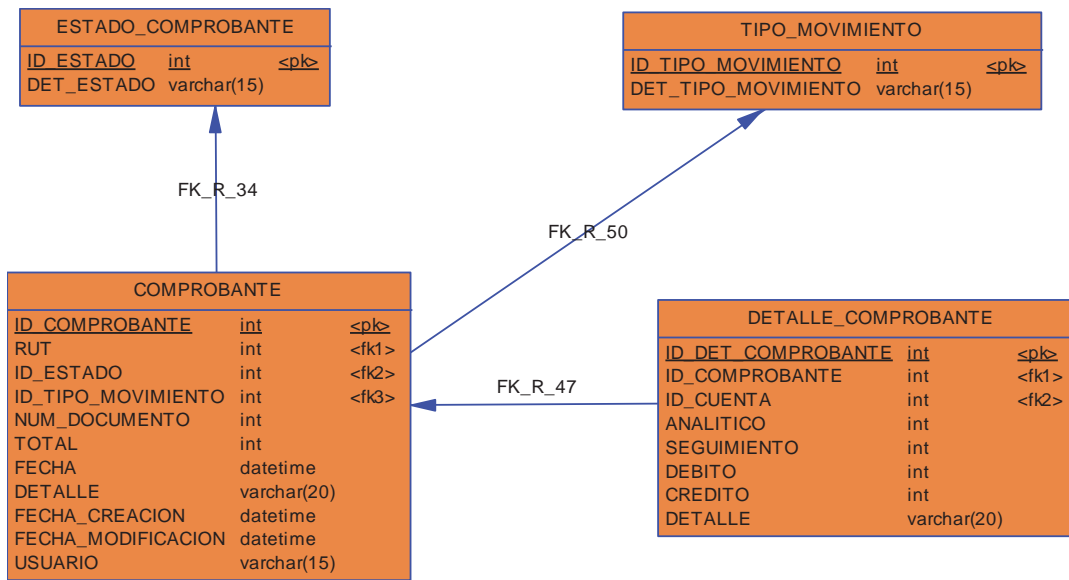


Figura 9.5: Sección relativa a COMPROBANTE

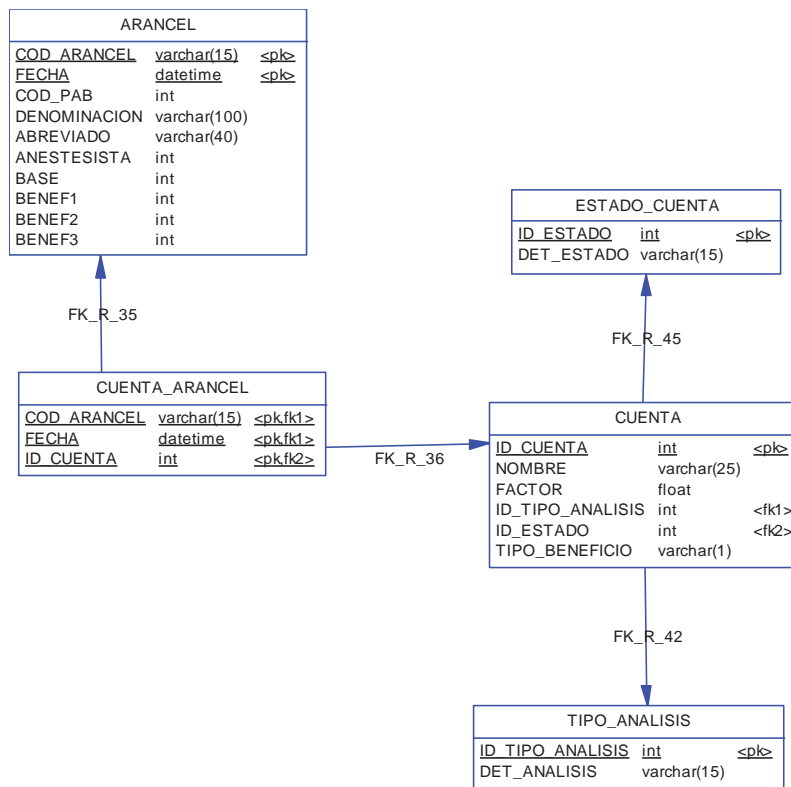


Figura 9.6: Sección relativa a CUENTAS CONTABLES

G: Modelo de Datos Segunda Iteración

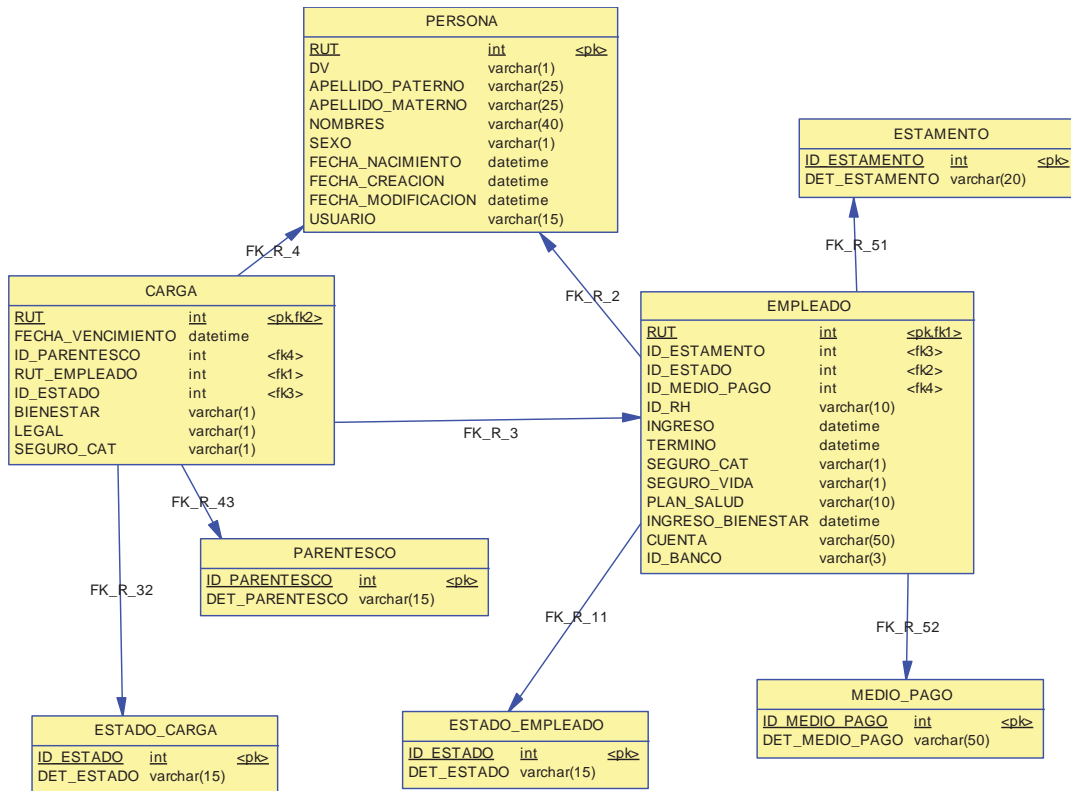


Figura 9.7: Sección relativa a PERSONA.

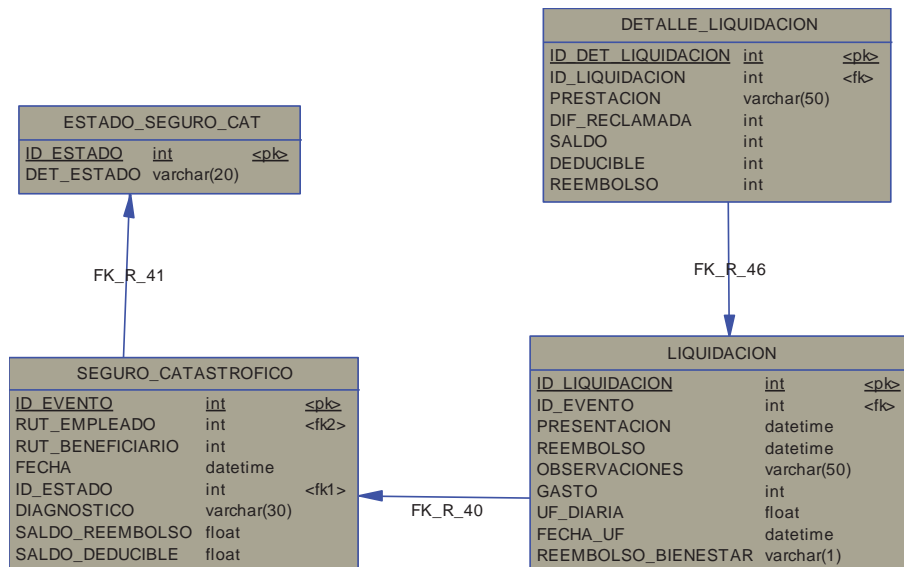


Figura 9.8: Sección relativa a SEGURO CATASTROFICO.

H: Tabla de Pruebas

Tabla 9.16: Plan de pruebas.

Funcionalidad	Tipo de Prueba	Dato de prueba	Respuesta esperada	Respuesta sistema
1. Ingresar Solicitud	dato válido	Folio de solicitud: 10	pasar al campo SOCIO	pasa al campo Rut socio
		Rut socio: 6992663	sistema complete el digito verificador y pase al campo COD	sistema responde como lo esperado
		COD: 6020 (correspondiente a recetas médicas)	sistema muestra el nombre correspondiente a la cuenta y salta al campo PAGADO	sistema responde como lo esperado
	dato no válido - Tipo de dato valido (numérico)	Folio de solicitud: -1	Sistema rechaza el dato ingresado, borrando el contenido del campo.	sistema responde como lo esperado
		Rut socio: 10	Sistema rechaza el dato ingresado, borrando el contenido del campo.	sistema responde como lo esperado
		COD: 1010 (no corresponde a alguna cuenta contable actualmente)	Sistema rechaza el dato ingresado, borrando el contenido del campo y mostrando un mensaje.	sistema responde como lo esperado
	dato no válido - Tipo de dato no válido (letra)	Folio de solicitud: 'asd'	Sistema rechaza el dato ingresado, borrando el contenido del campo.	sistema responde como lo esperado
		Rut socio: 'asd'	Sistema rechaza el dato ingresado, borrando el contenido del campo.	sistema responde como lo esperado
		monto pagado: 'asd'	Sistema rechaza el dato ingresado, borrando el contenido del campo.	sistema responde como lo esperado
	2. Buscar Solicitud	dato válido	búsqueda por folio: 10	Sistema debe mostrar los datos de la solicitud asociada al Folio.

		búsqueda por Rut: 16754361	sistema debe mostrar los datos de las solicitudes asociadas al Rut	sistema responde como lo esperado
	dato no válido - Tipo de dato valido (numérico)	búsqueda por folio: -1	Sistema debe mostrar información de solicitud, ya que no es válido el número de folio se muestra un mensaje de no encontrado.	sistema responde como lo esperado
		búsqueda por Rut: 10	sistema debe mostrar un mensaje señalando que el Rut ingresado no está registrado como socio	sistema responde como lo esperado
	dato no válido - Tipo de dato no válido (letra)	búsqueda por folio: 'asd'	Sistema debe rechazar la búsqueda.	El sistema borra el campo al ser un dato no válido.
		búsqueda por Rut: 'asdsd'	Sistema debe rechazar la búsqueda.	No realiza búsqueda ya que el dato no es válido.
3. Generar Comprobantes	Funcionalidad no permite datos de entrada, prueba se realiza en base a operar como corresponde	Ejecutar la generación de comprobantes.	El sistema debe cambiar las solicitudes que están en estado pendiente, pasando a estar aceptadas, y generar los comprobantes contables asociados en estado de pendientes.	sistema responde como lo esperado
4. Buscar Comprobantes	dato válido	búsqueda por folio: 10	El sistema debe mostrar el comprobante asociado a esa solicitud.	sistema responde como lo esperado
		búsqueda por Rut: 6992663	El sistema debe mostrar los comprobantes asociados al Rut del socio.	sistema responde como lo esperado
		búsqueda por comprobante: 1	El sistema debe mostrar el comprobante con ese número de registro.	sistema responde como lo esperado

4. Buscar Comprobantes	dato no válido - Tipo de dato valido (numérico)	búsqueda por folio: -1	El sistema debe mostrar un mensaje indicando dato no válido	sistema responde como lo esperado
		búsqueda por comprobante: -1	El sistema debe mostrar un mensaje indicando dato no válido	sistema responde como lo esperado
	dato no válido - Tipo de dato no válido (letra)	búsqueda por folio: 'asd'	El sistema debe rechazar la búsqueda por dato no válido	El sistema borra el campo al ser un dato no válido.
		búsqueda por Rut: 'asdsd'	El sistema debe rechazar la búsqueda por dato no válido	El sistema borra el campo al ser un dato no válido.
5. Manejo de Archivos -> Generar Depósitos	dato válido	Se selecciona un archivo con extensión (.xls), que será utilizado para escribir en él.	El sistema debe vaciar el archivo cargado y completarlo con los datos referentes a las cuentas a depositar y los beneficios.	sistema responde como lo esperado
	dato no válido	se selecciona un archivo con otra extensión	El sistema debe rechazar el archivo seleccionado.	sistema muestra un mensaje donde indica que el tipo de archivo no es permitido
6. Libros Contables	libro diario - mes con movimiento	Selecciono mes de Agosto 2013	Sistema debe mostrar datos junto a total.	sistema responde como lo esperado
	libro diario - mes sin movimiento	Selecciono mes de Enero 2013	Sistema debe mostrar total del mes anterior solamente.	sistema responde como lo esperado
	libro mayor - cuenta con movimiento	Selecciono mes de Agosto 2013	Sistema muestra todos los movimientos asociados a la cuenta seleccionada.	sistema responde como lo esperado
	libro mayor - cuenta sin movimiento	Selecciono mes de Enero 2013	Sistema muestra la cuenta con total mensual en cero.	sistema responde como lo esperado

6. Libros Contables	balance - mes con movimiento	Selecciono mes de Agosto 2013	Sistema debe mostrar el balance de 8 columnas junto al resultado del ejercicio.	sistema responde como lo esperado
	balance - mes sin movimiento	Selecciono mes de Enero 2013	Sistema debe mostrar el balance en 0 en todos los totales.	sistema responde como lo esperado
1. Ingresar Comprobante	dato válido	COD: 6020 (correspondiente a recetas médicas)	sistema muestra el nombre correspondiente a la cuenta y salta al campo ANALITICO	sistema responde como lo esperado
		ANALITICO(Rut Socio): 6992663 - jefe Bienestar Oscar Avila	El sistema debe agregar el nombre del socio en el campo detalle, y pasar al campo SEGUIMIENTO.	sistema responde como lo esperado
		SEGUIMIENTO: 0 (numérico)	El sistema debe pasar al campo DEBITO	sistema responde como lo esperado
	dato no válido (numérico)	COD: 1010 (no corresponde a alguna cuenta contable actualmente)	Sistema debe rechazar el dato ingresado, borrando el contenido del campo y mostrando un mensaje.	Sistema muestra un mensaje indicando que el dato no es válido.
		DEBITO: -1	El sistema debe rechazar el tipo de dato y mostrar un mensaje.	sistema responde como lo esperado
	dato no válido - Tipo de dato no válido (letra)	COD: 'asdas'	El sistema debe rechazar el tipo de dato y mostrar un mensaje.	sistema responde como lo esperado
		ANALITICO(Rut Socio): 'asdasd'	El sistema debe rechazar el tipo de dato y mostrar un mensaje.	sistema responde como lo esperado
2. Manejo de Archivos-> Generar CONTA05	dato válido	Se selecciona un archivo con extensión (.xls), que será utilizado para escribir en él.	El sistema debe vaciar el archivo cargado y completarlo con las columnas del archivo CONTA05 y los datos necesarios para llenar los cheques.	sistema responde como lo esperado

	dato no válido	se selecciona un archivo con otra extensión	El sistema debe rechazar el archivo seleccionado.	Sistema muestra un mensaje donde indica que el tipo de archivo no es permitido.
--	----------------	---	---	---